NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

(NASA-CR-134064) SYNTHETIC APERTURE RADAR AND DIGITAL PROCESSING (Kansas Univ.) 285 p

N73-73951

Unclas
00/99 18548

SYNTHETIC APERTURE RADAR AND DIGITAL PROCESSING

TECHNICAL REPORT 177-10 OF THE CENTER FOR RESEARCH, INC., ENGINEERING SCIENCE DIVISION, UNIVERSITY OF KANSAS
SUPPORTED BY NASA CONTRACT NAS 9-10261

MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

286

NOTICE

CRES CENTER FOR RESEARCH, INC. · ENGINEERING SCIENCE DIVISION
THE UNIVERSITY OF KANSAS · LAWRENCE, KANSAS · 66044

SYNTHETIC APERTURE RADAR AND DIGITAL PROCESSING

Technical Report 177-10

Ralph Gerchberg

September 1970

Supported by NASA Contract No. NAS9-10261

//

ACKNOWLEDGEMENT

The author owes a large debt of gratitude to Dr. R. K. Moore for suggesting the topic for study and for his guidance and counsel during the conduct of the research. It is fair to state that this project could not have been accomplished without his aid and encouragement.

Mr. Bruce Morgan, who handled the computer programming chores with skill and adroitness, deserves a large measure of thanks for his efforts. The versatility of the software which he helped to generate is reflected in the scope of the investigations carried out.

The support of the National Aeronautics and Space Administration under contract NAS 9-10261 is gratefully acknowledged.

The secretarial assistance which was so generously bestowed by Mary Bernier, Donna Opperman and Kay Nugent who persisted through several revisions of the manuscript is greatly appreciated.

Lastly, I would like to thank my beloved wife Louise, who did what was needed when it counted.

CONTENTS

Preceding page blank

CONTENTS

TABLES

FIGURES

FIGURES

INTRODUCTION TO THE PROBLEM

This is a study of the operation of Synthetic Aperture Radar (SAR) systems for generating radar images of terrain. The study is made with the purpose in mind of determining whether such an electronic system could be built, using digital techniques, to image in real time. Although the literature abounds with information on the optical processor[1,2,3,4,5,6] for generating SAR images, little has been published on the possibility of real time imaging with or without digital techniques.

The dominant motivation for performing this study is the idea of instrumenting an earth orbiting platform with a real time SAR imager for monitoring earth resources. Therefore, a situation which will be used throughout this paper as an illustrative example will be that the SAR is carried aboard a spacecraft "flying" in a circular orbit at an altitude of 600 nautical miles. This means that the ground speed of the satellite is approximately 7 kilometers per second. It will be assumed that the ground map that is to be generated by the orbiting radar will have a 30 meter resolution and will image a swath (parallel to the satellite track) 40 kilometers wide. Schwarz, Simonett, Jenks, and Ratzlaff[7] have discussed the problem of resolution in terrain imagery from space and it is clear that 30 meters is a useful resolution for thematic land use maps. The SAR will be a side-looking radar. The geometry of the imaging system is pictured in Figure 1.

For purposes of the example, the physical antenna length in the along-track direction will be taken to be 4 meters. It would be difficult to achieve a larger antenna length aboard a platform of say the NIMBUS satellite class. However, larger platforms could conceivably carry proportionately larger antennas. This constraint bears on the required system pulse rate to record the complete doppler history of a target unambiguously. Hence, very much in line with Harger's[6] words, it sets the speed at which the digital processor must work,

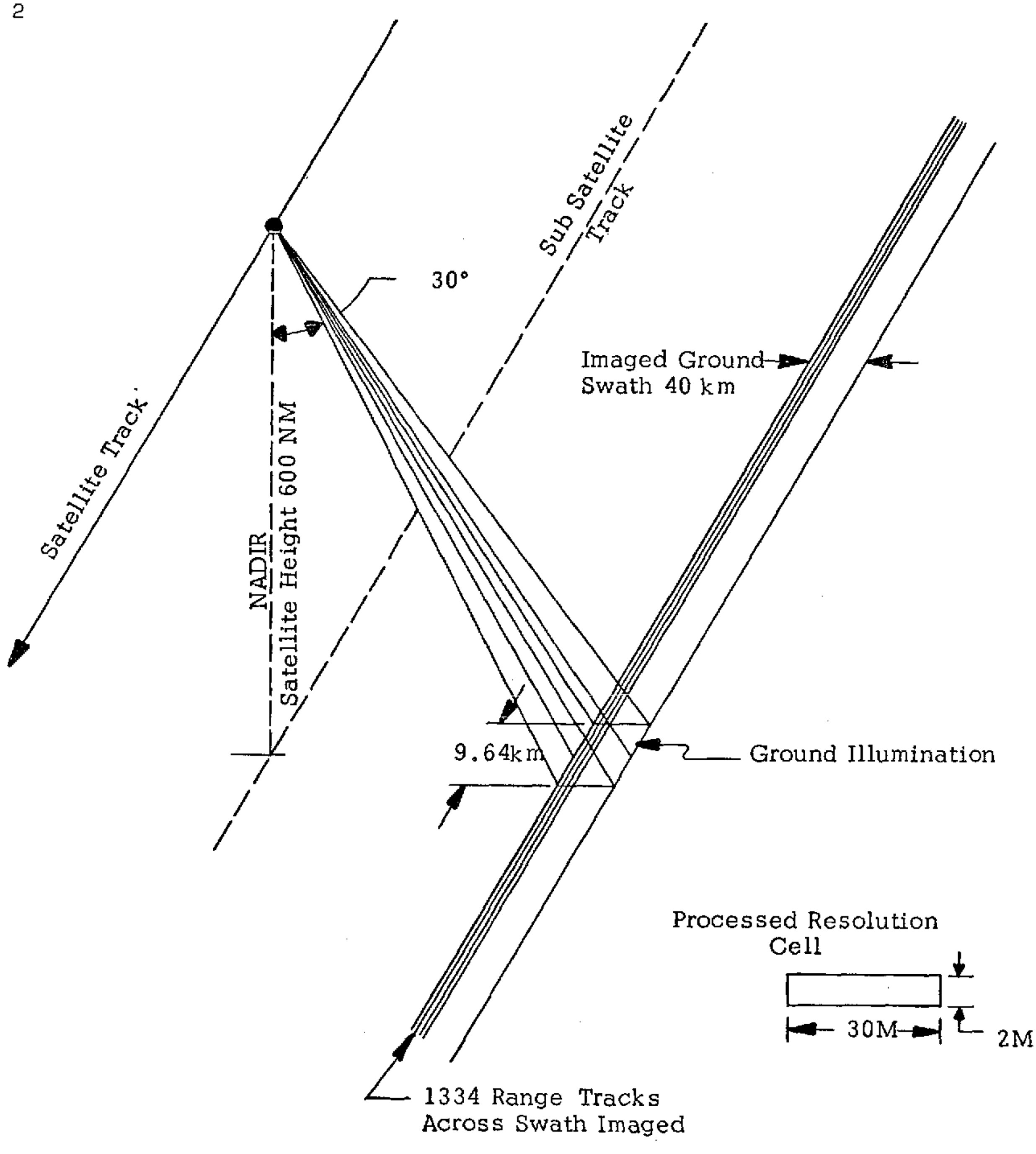


Figure 1. Side looking synthetic aperture terrain imaging geometry situation parameters: satellite altitude 600 nautical miles, ground swath imaged is 40 kilometers wide, rectangular beam assumed. The physical antenna along-track length is 4 meters, and the pulse duration is 0.1 μ seconds.

so that the system will generate an image in real time. It is readily shown, via Nyquist's[8] sampling theorem, that the minimum PRF is such that there be 2 pulses per translation of the physical antenna length along the satellite track. For a 4 meter long antenna, this means that the minimum PRF is 3500 Hz.

According to Figure 1 the physical antenna will be bore-sighted 30 degrees off the nadir and normal to the platform's velocity vector. Thirty meter cross track resolution determines that the pulse duration be 0.10 μ-seconds. Therefore, the maximum spatial frequency in the image cross track will be 1/60 cycles per meter and by the Nyquist criterion the image is sampled at least once every 30 meters. In other words, the number of image lines which must be generated across the swath imaged is given by 40,000/30 or 1334. The image is said to be divided into 1334 nominal range bins in the cross-track direction.

The 4 meter along-track physical antenna length sets the physical beam angle in along track direction (assuming a carrier frequency of 10 GHz) at 0.42 degrees making the nominal along-track distance illuminated by the physical antenna beam on the ground 9.64 kilometers. The doppler bandwidth of the return signal is $2v/\ell$ (v is the platform velocity= 7×10^3 meters per second; ℓ is physical antenna length along track = 4 meters) or 3500 Hz.

A very important parameter is the time-bandwidth product (TB) where T is the time for the physical beam to fly by a point target on the ground and B is the doppler bandwidth. This example yields a time-bandwidth (TB) product of 4820. In Chapter II, it is shown that the return signal from any point, is a sampled (at the PRF), linearly swept, F. M. signal whose bandwidth is B and whose duration is T. As a direct consequence, "chirp" techniques[9] developed for pulse compression radar systems can be used to analyze this method for fine along-track resolution. In fact, this is the reason why Brown[10] classes synthetic aperture radar with pulse compression techniques.

The SAR system studied here employs a homodyne technique so that the output of the receiver is translated to a zero carrier frequency. An optimum algorithm is shown in Chapter II which requires the processor to use quadrature coherently detected return signals. The processor must store in each range bin, for the example situation, the last 4820 returns or 9640 numbers (with quadrature detection) for correlation at any instant to achieve Full Focused processing. The correlation process requires 19280 products to be formed in a time period of the $(PRF)^{-1}$ in each range bin.

In Chapter III, variations on the processing algorithm are considered. Among the techniques which are discussed is one which the author dubs "subaperturing." In effect, the technique allows for focussed and zone plate processing on contiguous parts of the doppler return spectrum to create a series of synthetic subapertures all squinted at different angles within the physical beam width and all non-overlapping. This technique allows one to process for the along-track resolution desired (30 meters in the example) rather than for the best along-track resolution possible (2 meters in the example). As a result, the size of storage required in the processor is reduced by the image resolution degradation factor. In the example, the 9640 numbers per range bin becomes 642 numbers. A further investigation is made into the nature of the processor algorithm and it is shown that for a price, an image may be generated using non-quadrature detection and only one cross-correlation. Therefore, under this option 75 percent of the computation involved in the processing algorithm is eliminated. The price exacted for this truly remarkable saving in storage and computation time is in terms of image quality -- in gray tone resolution.

In Chapter III, a figure of merit for the radar image of terrain, the mean to standard deviation ratio (M/STD) is adopted. The point is made that under perfect imaging of terrain, employing a Rayleigh model, the best image M/STD ratio achievable is 5.61 db. With non-quadrature processing, the best image will have an M/STD ratio 3 db lower. However, one may sum the images created by multiple subapertures to enhance the image M/STD ratio. The enhancement is in direct proportion to the square root of the number of subapertures used (the maximum number of subapertures in the example is 15). Processor computation increases in direct proportion to the number of subapertures used. It turns out that the storage per range bin also increases with the number of subapertures used. And, in fact, the number of words stored just doubles if all possible subapertures are used. Interestingly though, the additional stores are approximately 2 orders of magnitude slower than the primary stores which supply the numbers for cross-correlation. This suggests that the two different kinds of stores may employ two distinct technologies.

A large computer program, carefully documented in Appendix III, has been created to simulate the SAR system and the various versions of processing discussed in Chapter III. Chapter IV ties the input and output data to and from the program to the physical phenomena which are being simulated. Included in the program are the effects of I.F. gain non-linearities, antenna pattern, system noise, doppler mismatch, quantization, presumming, etc., as well as a host of diverse processing schemes.

Chapter V presents a considerable amount of data in graphical form, documenting the effects which various SAR parameters have on the radar images generated. One figure of the Chapter, Figure 29a, page 127, is extremely significant in that it shows the effect of M/STD on a simulated radar image. It also exemplifies the way in which subaperture processing enhances M/STD.

Chapter VI discusses the present state of technology with regard to realizing an SAR digital processor. The thrust of Chapter VI is that the only technology which can be used to build such a processor is the rapidly burgeoning Large Scale Integrated circuit (LSI) technology. A table is presented in this Chapter which makes it possible to readily figure the storage per range bin required for non-quadrature subaperture processing. It is shown that to attain a M/STD of $\sqrt{18}$, a storage of 4.75 megabits is required in a typical space application. This, of course, may be diminished by sacrificing image swath width and/or M/STD. Using Khambata's[11] idea of a multi-processor and assuming that all range bins can be processed in parallel, though each range bin's returns are processed serially within the bin, the time for a multiplication in 1.3 μ seconds in the example situation. This time is relatively slow for LSI technology. The packaging density is so high under this technology that 4.75 megabits need not be overly large. However, the technology is so new that one has no large main frame computer memories as of this writing to show as an example of a working system. There is the problem of power dissipation which is cited. The relatively well established methods employed in LSI require considerable power. In one instance, 4.75 megabit memory now in design and expected to be marketed in late 1971 would consume something over two kilowatts. However, much research and development is going on in this area and all reports indicate that "complimentary metallic oxide semi conductors" (CMOS) large scale integrated circuits will be available within the next few years. They will have power dissipations of the order of a few microwatts per bit. These devices are still in the future. Based on anticipated 1975 LSI technology, an SAR design is given employing real time digital processing.

The conclusion presented in Chapter VII is that for <u>extremely</u> limited SAR images a digital processor could be built today employing LSI techniques now well established. For excellent radar imagery the size of the processor, storage-wise, requires tens of kilowatts to operate (assuming 30 meter resolution, an image M/STD of $\sqrt{18}$ and a 40 kilometer ground swath). However, by 1975, LSI-CMOS technology, which is just beginning to develop now, should cut the processor storage power requirement to something below 100 watts, making a digital processor readily achievable.

BASIC PRINCIPLES OF SYNTHETIC APERTURE RADAR

Synopsis

This chapter contains background material on the principles of side-looking synthetic aperture imaging radar systems. The geometry of the radar imaging situation is discussed and it is shown that the return from a point target is a linearly swept F. M. signal; a chirp. The duration of the chirp (T) is the time the point target is in the physical antenna's beam and the bandwidth of the chirp (B) is the doppler frequency spread in the return signal.

It is established that in general there is a minimum time-bandwidth (TB) product for a signal (of the order of one) and that this is achieved only when the signal phase in time (ϕ) or the spectral phase in the frequency domain(Θ) is linear or constant. This idea is used to achieve the fine along-track resolution characteristic of synthetic aperture radar. The signal returned in each range bin is cross correlated with the point target response, removing the non-linear spectral phase from the point target return in the frequency domain and compressing the signal duration in the time domain by (TB).

For the chirp signal the fact that the group delay is approximately constant leads to signal envelopes in the time domain being closely mimicked in the frequency domain. A technique to reduce subsidiary lobes in the point target response of the syntheitc aperture radar involves weighting the spectrum of the cross correlation reference function. This weighting can be done in the time domain because of the mimicking property.

Quadrature processing and the imaging algorithm are discussed and it is shown that for zero offset frequency, quadrature processing is necessary. Otherwise, targets with reflection coefficients with random phases create a distorted image.

The concept of a radar ambiguity diagram for a synthetic aperture radar is developed. It is shown that the usual type of ambiguity diagram showing system response as a function of delay and frequency mismatch is still applicable with modification to include radial acceleration of the target.

Geometry of SLR Imaging

The geometry of a side-looking imaging radar system is shown in Figure 1. As the platform bearing the radar system moves along the flight path, it illuminates the swath to be imaged with pulses of radiation. The bandwidth of the pulse determines the nominal resolution of the system in the cross-track direction and without signal processing, the azimuth or along-track beam width of the antenna determines the nominal along-track resolution. Cross-track resolution is usually achieved with a short amplitude modulated pulse or a longer pulse with suitable modulation to permit pulse compression. The most widely used pulse compression waveform is a linearly swept frequency modulated signal. Figure 2 shows these range resolution waveforms and their spectra. The time domain pulse shapes shown in the Figure are idealizations. A short duration pulse in time requires a very wide bandwidth system to pass largely undistorted. Since the nominal system bandwidth is finite, by the inverse character of time and corresponding frequency domain signals, the pulse must be infinitely long in time if it is to be totally undistorted by the system. Of course the solution to this problem, which might lead one to expect non-causal systems, is that while the bandwidth of the transmitted pulse is nominally finite, it does have negligibly small contributions across the entire spectrum. However, distortions in the spectrum due to the system filtering action (i.e., finite bandwidth transfer function) cause a broadening and rounding of the pulse as shown in Figure 3 and as well the presence of time side-lobes. These time side-lobes exist in all real systems though their magnitude may be altered by proper shaping of the system transfer

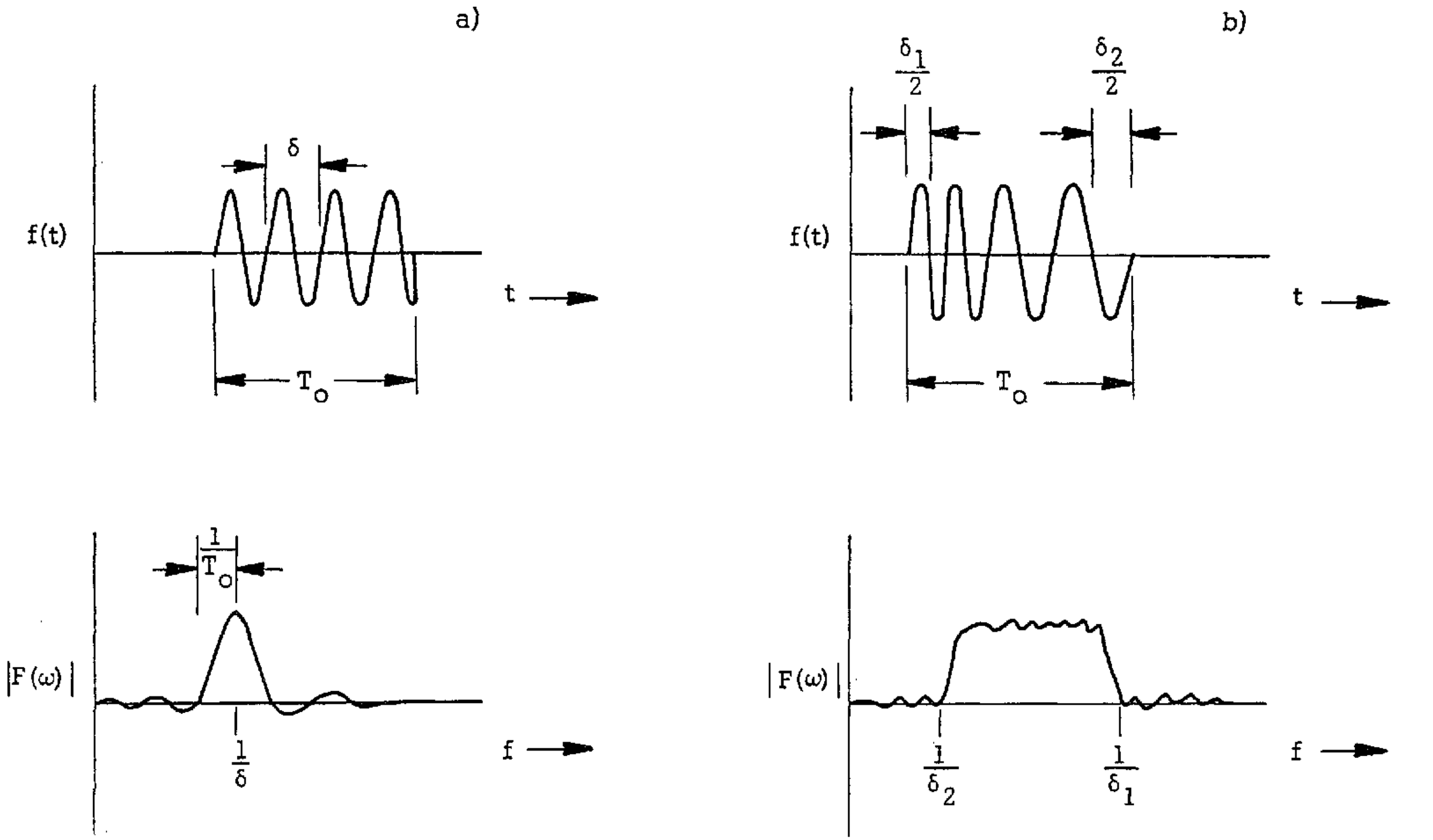


Figure 2. Two kinds of range resolution waveforms and their spectra:
a) constant carrier pulse; b) a linearly swept FM pulse.

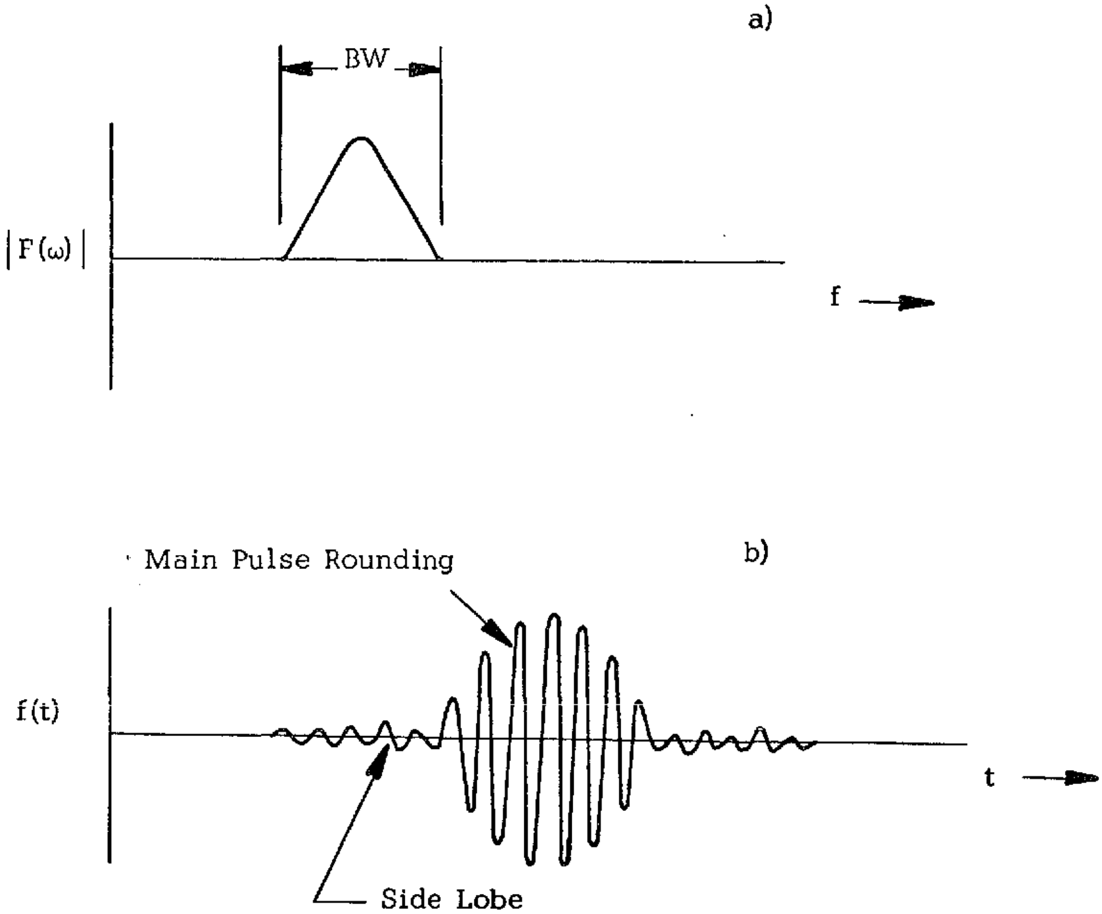


Figure 3. The result of band-pass filtering the amplitude modulated pulse shown in Figure 2a. a) The waveform spectrum; b) The time waveform.

function. They represent ambiguities in the cross-track or slant-range position of a target in the vicinity of strong targets or they may be interpreted as weak targets when there are none, in the vicinity of a single strong target.

Resolution much finer than the length of ground intercepted by the physical antenna's beam in the along-track direction is achievable by properly processing the signals returned during a time which is equal to that required to fly the distance on the ground illuminated by the physical beam. The processing makes use of the fact that for reasonably narrow azimuth beams ($<10^\circ$) the signal from a point target, assuming C.W. illumination, is a long pulse which is frequency modulated in a linear fashion. The fact that the target return is a train of pulses at the system PRF rather than a continuous signal is not important to the argument in this instance, since it may be thought of as a series of periodic samples from the C.W. signal generated by the point target. All that is required is to sample at a sufficiently high frequency (PRF) to avoid Nyquist rate ambiguity difficulties. Referring to Figure 4, assuming constant illumination intensity across the beam (i.e., a flat antenna gain in the along-track direction), and disregarding the slight amplitude change in the return signal due to the changing slant range as the beam is swept by the point target, the voltage return as a function of the radar position (x) is given by

$$V(x,t) = A\, e^{j2\pi f_0 t}\, e^{-j\frac{4\pi}{\lambda}\ell}\, e^{j\phi_0}$$

Here ϕ_0 is an arbitrary phase factor stemming from the complex reflection coefficient associated with the point target.

but $$\ell = R^2+x^2 \approx R\left(1+\frac{x^2}{2R^2}\right)$$

or $$\ell \approx R + \frac{x^2}{2R}$$

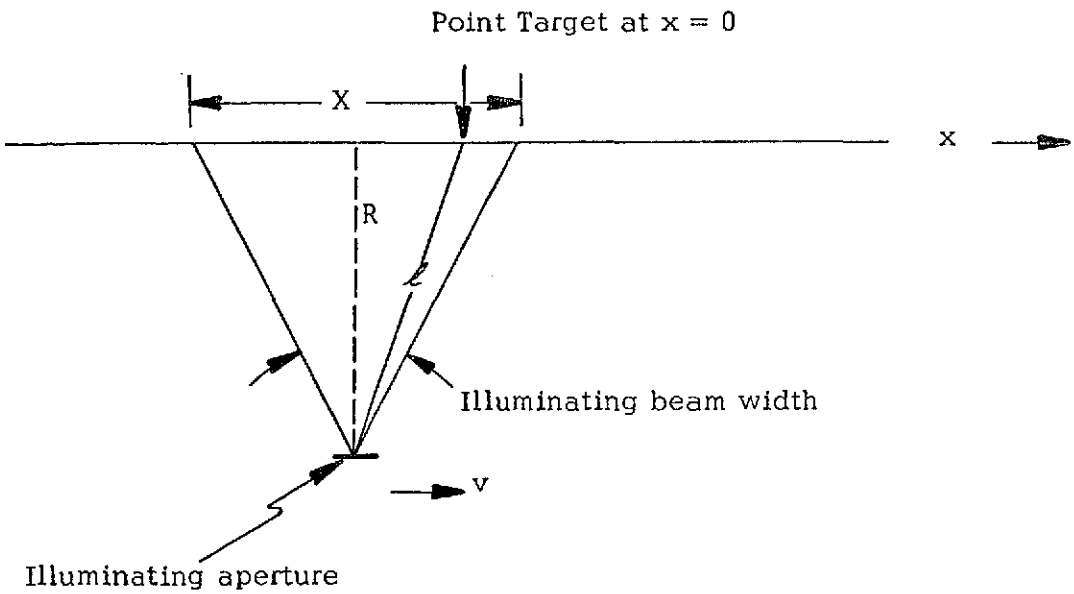


Figure 4. Point target imaging geometry by side-looking synthetic aperture radar.

thus $V(x,t) = A \,\mathrm{rect}\left(\frac{x}{X}\right) e^{j\left(-\frac{4\pi}{\lambda}R + \phi_0 + 2\pi f_0 t\right)} e^{-j\frac{2\pi}{\lambda R}x^2}$

where $\mathrm{rect}\left(\frac{x}{X}\right) = \begin{cases} 1 & \text{for } |x| \le \frac{X}{2} \\ 0 & \text{for } |x| > \frac{X}{2} \end{cases}$

This "chirp" signal's instantaneous frequency is a linear function of time or distance. The instantaneous frequency is a characteristic which has a clear physical meaning for a signal with a relatively small percentage bandwidth. It means that the signal within a small time increment about the instant of concern is well approximated by a sinusoid of the instantaneous frequency f_i. Thus for the chirp pulse given above:

$$f_i = \frac{1}{2\pi}\frac{d}{dt}\phi(t) = \frac{1}{2\pi}\frac{d}{dt}\left(-\frac{4\pi}{\lambda}R + \phi_0 + 2\pi\left(f_0 t - \frac{2\pi}{\lambda R}v^2t^2\right)\right)$$

$$f_i = f_0 - \frac{2v^2}{\lambda R}t$$

where v is the along-track velocity of the radar system (see Figure 4). To say that there is an extensive body of literature on the lore of the chirp signal would be a gross understatement. Perhaps the classic article on chirp was that written by Klauder, et al[9], in 1955. The chirp waveform has certain properties which bear on the system processor design. However, before these can be fully appreciated, it is necessary to study properties of pulse compression type waveforms generally.

Time-Bandwidth Product

Matched filtering is the touchstone allowing synthetic aperture processing to achieve along-track resolutions finer than that provided by the physical antenna's along-track beam width. The reason for this becomes evident when one considers the nature of a waveform of finite duration. Associated with this waveform is a nominal bandwidth containing better than say 95% of the waveform energy. It may be that the energy of the waveform is distributed in several separated frequency bands. In this case the effective bandwidth would be the sum of the bandwidths of the sub-bands. One can show that the time bandwidth product of the waveform has a minimum limit of the order of one, depending on the way that one defines the bandwidth. Starting with a waveform of large time-bandwidth product then, it appears possible via processing to reduce the waveform's duration or its bandwidth by the inverse of its bandwidth or duration, respectively. It is, of course, also possible to reduce both duration and bandwidth simultaneously. No claim is made here regarding the efficiency of the compression process, vis-a-vis the percentage energy of the uncompressed pulse which is contained in the nominal duration of the compressed pulse, but it can be shown that for matched filtering the conversion efficiency is almost 100% and that the minimum time-bandwidth product is attained.

The carrier signal is defined, using exponential representation,[12] so that the mean of the energy density spectrum is zero about the carrier frequency (f_o).

$$s(t) = a(t)\cos(2\pi f_o t + \phi(t))$$

but $\mathcal{F}[a(t)\, e^{j\phi(t)} e^{j2\pi f_o t}] = \mathcal{F}[a(t)e^{j\phi(t)}] * \mathcal{F}[e^{j2\pi f_o t}]$

defining $M(f) \triangleq \mathcal{F}[a(t)e^{j\phi(t)}]$

yields $0 = \int_{-\infty}^{\infty} f\,|M(f)|^2\, df$

The rms signal bandwidth is defined as the second moment of the modulation function energy spectrum.

$$\delta_f = \left[\int f^2 |M(f)|^2 df \Big/ 2E\right]^{1/2}$$

where E is the waveform energy

$$E = \int_{-\infty}^{\infty} s^2(t)\, dt = \frac{1}{2}\int_{-\infty}^{\infty} |M(f)|^2 df = \frac{1}{2}\int_{-\infty}^{\infty} |\psi(t)|^2 dt$$

Similarly, the rms time duration of the signal is based on the second moment of the modulation envelope squared.

$$\delta_t = \left[\int_{-\infty}^{\infty} t^2 |\psi(t)|^2 dt / 2E \right]^{1/2}$$

To establish a lower limit on the time bandwidth product of the waveform, first one establishes a general relation between the n^{th} derivative of a waveform, in either the time or frequency domain and its transform in the opposite domain. Then it can be shown that a waveform which has a linear or constant phase function with time (ϕ (t)), has the minimum bandwidth for a given waveform envelope (a(t)). Finally, Schwarz's inequality can be employed to fix the lower limit of the time-bandwidth product. It is realized that an appeal to the literature would be the most expeditious method of establishing the lower limit on the time-bandwidth product of a waveform. However, there is much to be gained in following the procedure outlined, particularly in the way of understanding the fundaments of compression waveforms.

Consider a function g(t) with Fourier transform (G(f)). Then the n^{th} derivative of g(t) will have the following transform:

$$\mathcal{F}\left[\frac{d^n}{dt^n} g(t)\right] = \int_{-\infty}^{\infty}\left[\frac{d^n}{dt^n}\int_{-\infty}^{\infty} G(u) e^{j2\pi ut} dt\right] e^{-j2\pi ft} dt$$

taking the derivative inside the integral

$$\mathcal{F}\left[\frac{d^n}{dt^n} g(t)\right] = (j2\pi)^n \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} u^n G(u) e^{j2\pi ut} du\right] e^{-j2\pi ft} dt$$

and appealing to Fubinis theorem we reverse the order of integration

$$\mathcal{F}\left[\frac{d^n}{dt^n} g(t)\right] = (j2\pi)^n \int_{-\infty}^{\infty} u^n G(u) \delta(f-u) du$$

or

$$\mathcal{F}\left[\frac{d^n}{dt^n} g(t)\right] = (j2\pi)^n f^n G(f)$$

Similarly

$$\mathcal{F}^{-1}\left[\frac{d^n}{df^n} G(f)\right] = (-j2\pi)^n t^n g(t)$$

Note that we have derived two relations between derivatives of a function in one domain and the moment of the transformed function in the inverse (Fourier transformed) domain. These relations may be put in a useful integral form with the aid of Parseval's theorem as follows:

Parseval's Theorem:

$$\int_{-\infty}^{\infty} u(x)v^*(x)\,dx = \int_{-\infty}^{\infty} U(f)V^*(f)\,df$$

$$(-j2\pi)^n \int_{-\infty}^{\infty} t^n |g(t)|^2 dt = \int_{-\infty}^{\infty} G^*(f) \frac{d^n}{df^n} G(f)\,df$$

$$(j2\pi)^n \int_{-\infty}^{\infty} f^n |G(f)|^2 df = \int_{-\infty}^{\infty} g^*(t) \frac{d^n}{dt^n} g(t)\,dt$$

Thus the rms bandwidth of the waveform and the rms duration may be written in alternate forms to those given previously as follows:

$$\delta_f^2 = \int_{-\infty}^{\infty} f^2 |M(f)|^2 df \Big/ 2E = \frac{1}{2E}\frac{1}{(j2\pi)^2}\int_{-\infty}^{\infty} a(t)e^{-j\phi(t)} \frac{d^2}{dt^2}\left(a(t)e^{j\phi(t)}\right)dt$$

Integrating by parts yields

$$\delta_f^2 = -\frac{1}{2E(j2\pi)^2}\int_{-\infty}^{\infty}\left\{(a'(t))^2 + (a(t)\,\phi'(t))^2\right\}dt$$

Hence the minimum rms bandwidth occurs with ϕ (t) equal to zero. ϕ'(t) cannot equal a constant as this would imply a carrier frequency where we have specified M (f) to be for the modulation only at zero carrier frequency. The conclusion is that any non-linear phase factor (ϕ(t)) in the modulating envelope increases the bandwidth of the envelope. The minimum rms bandwidth is achieved when ϕ'(t) is zero.

$$\delta_{f_{min}} = \frac{1}{\pi\sqrt{8E}}\left[\int_{-\infty}^{\infty}(a'(t))^2\,dt\right]^{1/2}$$

Similarly, the mean square duration of a waveform may be written as:

$$\delta_t^2 = \int_{-\infty}^{\infty} t^2 |\psi(t)|^2 dt/2E = \frac{1}{2E}\frac{1}{(-j2\pi)^2}\int_{-\infty}^{\infty} M^*(f)\frac{d^2}{df^2}M(f)\,df$$

By analogy to the mean square bandwidth treated above, it is clear that minimum duration is achieved, in a signal of a given spectral amplitude, if the associated phase as a function of frequency is constant or linear in time. Linear phase signifies a time delay only, with no distortion in the waveform.

The minimum time-bandwidth product may be derived as follows:

$$\delta_{f_{min}}^2 \delta_t^2 = \frac{1}{8\pi^2 E}\int_{-\infty}^{\infty} |a'(t)|^2 dt \; \frac{1}{2E}\int_{-\infty}^{\infty} t^2 |a(t)|^2 dt$$

Invoking Schwarz's inequality yields

$$\delta_{f_{min}}^2 \delta_t^2 \geq \frac{1}{16\pi^2 E^2}\left|\int_{-\infty}^{\infty} a(t)\; t\; a'(t)\; dt\right|^2$$

but

$$\int_{-\infty}^{\infty} a(t)\,t\,a'(t)\,dt = a^2(t)\,t\Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} a(t)\,t\,a'(t)\,dt - 2E$$

thus

$$\delta_f \delta_t \geq \frac{1}{4\pi}$$

In the process of deriving the minimum time-bandwidth product of a waveform two very important principles were uncovered, namely that the minimum bandwidth pulse has no non-linear phase term ($\phi(t)$) in its modulation, and that the minimum rms signal duration for a given amplitude spectrum is achieved when the associated phase function of frequency ($\Theta(f)$) in the frequency domain is linear or constant.

The Linearly F. M. Modulated Pulse Compression Signal

It has been shown that the signal received by a side-looking imaging radar system from a point target (assuming C.W. illumination) is of the form:

$$s(t) \propto a(t)\, e^{j\left(2\pi f_c t - \frac{2\pi}{\lambda R} v^2 t^2 + \phi_m\right)}$$

where $a(t)$ is the 2 way along track (azimuth) antenna gain

ϕ_m is an arbitrary phase angle.

In order to study the properties of this waveform, consider first an infinite time-bandwith product linearly swept F. M. signal:

$$s(t) = e^{j(2\pi f_0 t - \frac{2\pi}{\lambda R} v^2 t^2 + \phi_n)}$$

The Fourier Transform of this signal is determined as follows:

Suppose

$$s(t) = e^{-jkt^2}$$

$$S(f) = \int_{-\infty}^{\infty} e^{-j(2\pi f t + kt^2)} dt$$

$$= \int_{-\infty}^{\infty} e^{-jk(t + \pi f/k)^2} e^{j\pi^2 f^2/k} dt$$

$$= 2e^{j\pi^2 f^2/k} \left[\int_{-\pi f/k}^{\infty} \cos k(t - \tfrac{\pi f}{k})^2 dt - j\int_{-\pi f/k}^{\infty} \sin k(t - \tfrac{\pi f}{k})^2 dt \right]$$

With a change of variables such that

$$\frac{\pi}{2}\mu^2 = k(t - \pi f/k)^2$$

$$S(f) = 2e^{j\pi^2 f^2/k} \sqrt{\frac{\pi}{2k}} \left[\int_0^{\infty} \cos \frac{\pi}{2}\mu^2 d\mu - j\int_0^{\infty} \sin \frac{\pi}{2}\mu^2 d\mu \right]$$

The integrals in the brackets are recognized as Fresnel integrals and finally we have

$$S(f) = \sqrt{\frac{\pi}{k}}\ e^{j\pi^2 f^2/k}\ e^{-j\pi/4} \Longleftrightarrow s(t) = e^{-jkt^2}$$

However, the infinite time-bandwidth signal whose transform we seek has a carrier frequency (f_0) and an arbitrary phase factor (ϕ_n). Using the fact that multiplication in the time domain is the equivalent of convolution in the frequency domain, the transform of the signal is given as

$$s(t) = e^{j\left(2\pi f_0 t - \frac{2\pi}{\lambda_0 R} v^2 t^2 + \phi_n\right)} \Longleftrightarrow \frac{1}{v}\sqrt{\frac{\lambda_0 R}{2}}\, e^{j\frac{\pi \lambda_0 R}{2 v^2}(f - f_0)^2}\, e^{j\left(-\frac{\pi}{4} + \phi_n\right)}$$

The above equation demonstrates striking elements of symmetry between the infinite time-bandwidth product linearly swept F. M. wave in the time and frequency domains:

1. In both domains the amplitude is constant
2. In both domains there is a quadratic phase factor

There are still other dualisms which may be explored if one considers certain derivatives.

Examine the general relationship between phase , delay, and frequency in the time and frequency domains:

$$\mathcal{F}[f(t - t_0)] = e^{-j2\pi f t_0} F(f)$$

$$\mathcal{F}[e^{j2\pi f_0 t} f(t)] = F(f - f_0)$$

The dual of the instantaneous frequency is the group delay (τ_g) which is defined as

$$\tau_g = -\frac{1}{2\pi}\frac{d}{df}\,\theta(f)$$

where $\theta(f)$ is the signal's spectral phase function

For the infinite time-bandwidth product chirp signal, τ_g has physical significance. Consider any, say, 1 kilohertz increment in the infinite spectrum of the signal. The effective delay time for the energy contained in this frequency increment is given by τ_g according to the general Fourier Transform relations given above. Like the instantaneous frequency the concept of group delay (τ_g) is only approximate. But for signals with reasonably large time-bandwidth products, say greater than 10, it is often sufficient to associate the instantaneous waveform frequency with its delay time ($\tilde{\tau}_g$). The concept of group delay implies that the spectral phase function has a monotone increasing or decreasing slope. It is on this basis that one is able to say that the nominal bandwidth of a chirp signal is the difference of the instantaneous frequency at the beginning and ending of a chirp pulse.

The expression for τ_g was derived under the assumption of infinite time-bandwidth product. The question which still must be answered is to what extent does a finite time-bandwidth signal degrade the concept of group delay ($\tilde{\tau}_g$) and instantaneous frequency (f_i). A real signal has finite duration and bandwidth:

$$s(t) = a(t)\, e^{j(2\pi f_0 t - k t^2 + \phi_m)}$$

One would like to be able to determine the spectrum of this signal based on the infinite time-bandwidth spectrum in the following way:

$$f_i(t) = \frac{k}{\pi} t$$

$$s(t) = a(t)\, e^{j\phi_i}\, e^{-jkt^2}$$

$$S(f) = \sqrt{\frac{\pi}{k}}\, e^{-j\frac{\pi}{4}}\, a\!\left(\frac{\pi f}{k}\right) e^{j\phi_i}\, e^{-jk\left(\frac{\pi f}{k}\right)^2}$$

In other words, we have assumed a correspondence between instantaneous frequency and group delay based on the infinite time-bandwidth chirp spectrum derived above. Corresponding time and frequency domain waveforms involve a complex factor $\sqrt{\pi/k}\; e^{-j\pi/4}$ in the frequency domain and the substitution of $\pi f/k$ for t. This is a reasonably good approximation for waveforms with nominal time-bandwidth (as opposed to r m s time-bandwidth) products greater than 10. Figures 5 and 6 show the extent to which these approximations are valid for a normalized strict chirp pulse:

$$s(t) = \sqrt{\frac{1}{T}}\;\; rect\left(\frac{t}{T}\right) e^{-jkt^2}$$

Note that the nominal bandwidth (B) and the nominal pulse duration (T)

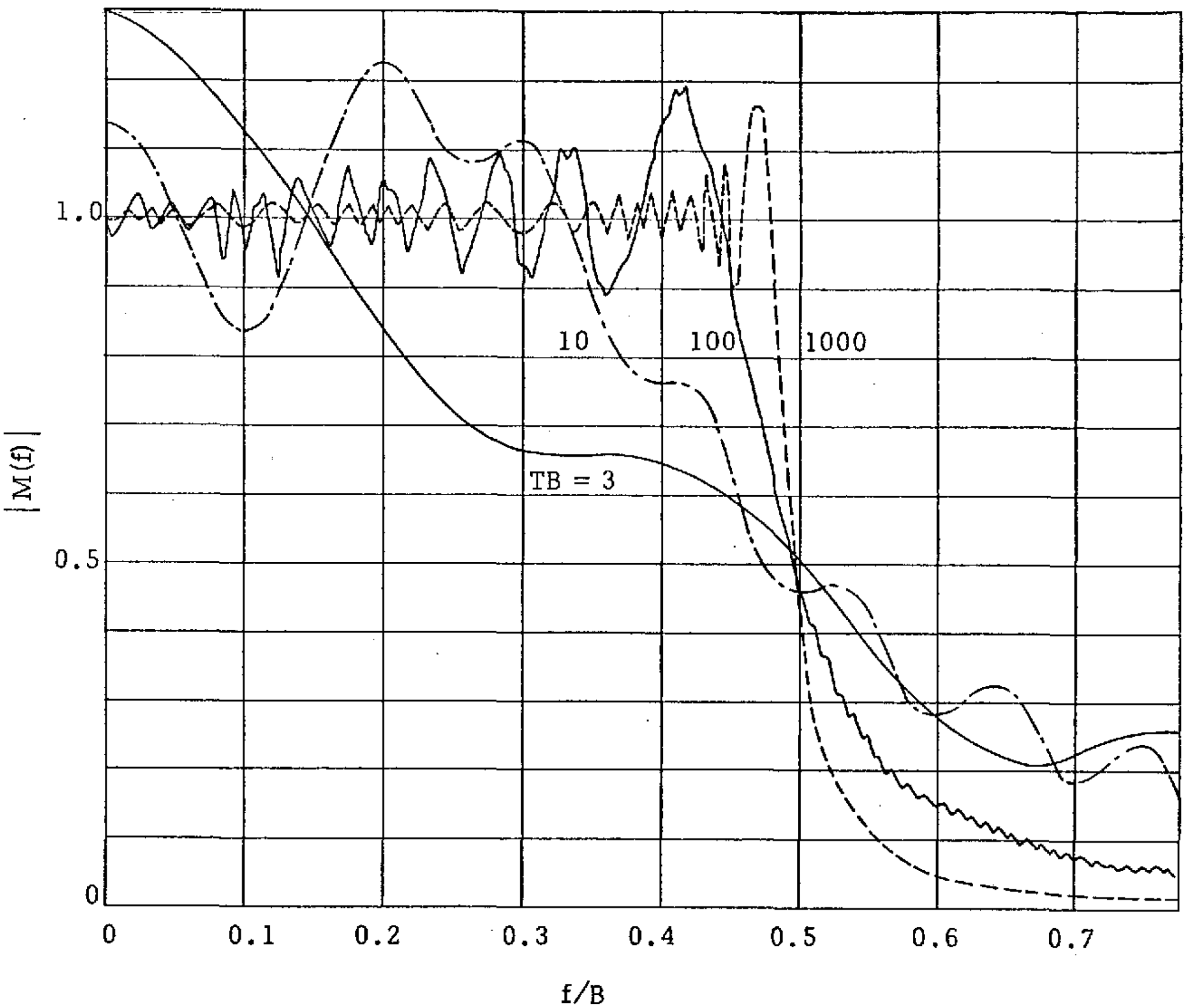


Figure 5. Frequency spectrum of a chirp signal for various values of TB (From Reference 13, page 232).

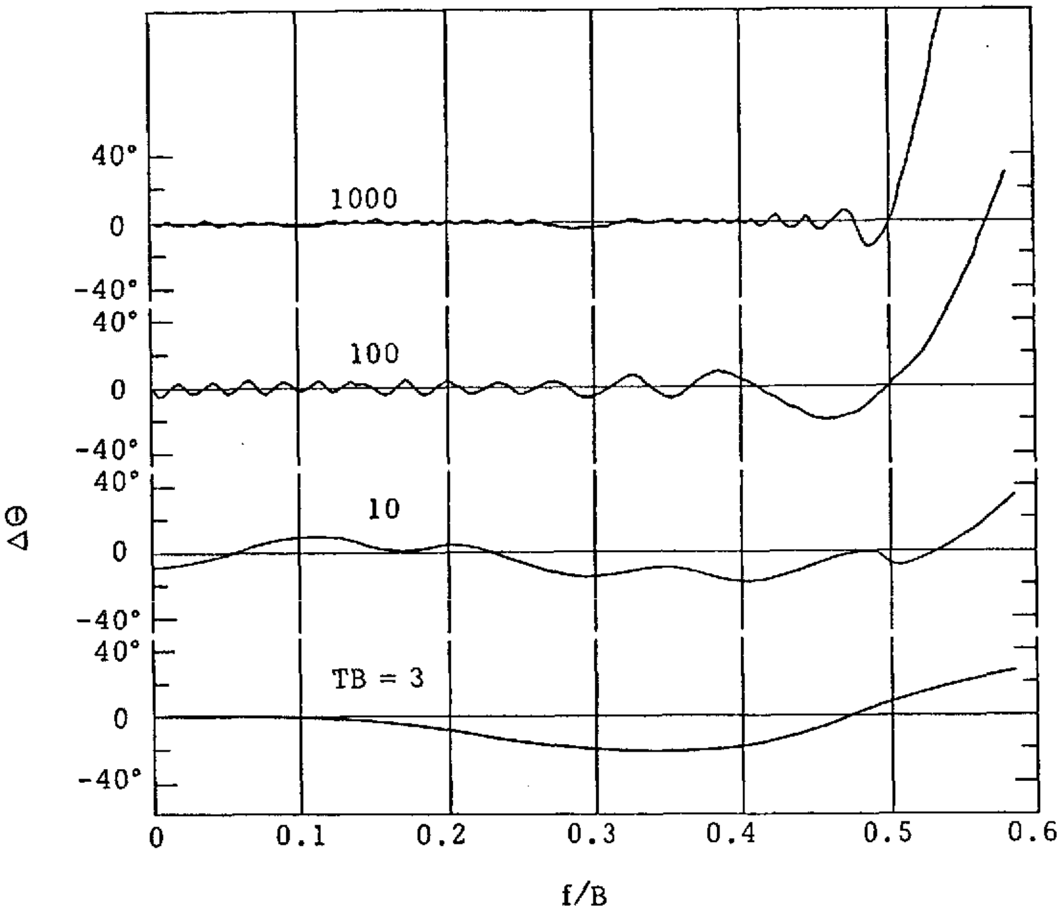


Figure 6. Error in the phase spectrum of the "large" TB product approximation $\Delta\Theta = \Theta_{Actual} - \left(-\frac{\pi}{4} - \frac{\pi f^2}{k}\right)$ (From Reference 13, page 233).

referred to in these figures are based respectively on the difference of instantaneous frequencies at the beginning and ending of the waveform and on the corresponding differences in group delays. Thus

$$B = \frac{k}{\pi}\left(t_{end} - t_{start}\right) \triangleq \frac{k}{\pi} T$$

The corresponding rms bandwidth (δ_f) and duration(δ_t) are equal respectively to ($B/\sqrt{12}$) and ($T/\sqrt{12}$).

As a matter of reference, the exact spectrum of the strict chirp waveform is

$$s(t) = rect\left(\frac{t}{T}\right) e^{j\phi_0} e^{-jkt^2}$$

$$S(f) = e^{j\pi(TB)\left(\frac{f}{B}\right)^2} e^{j\phi_0}$$

$$\left\{ C\left[\sqrt{2TB}\left(\frac{f}{B}+\frac{1}{2}\right)\right] - C\left[\sqrt{2TB}\left(\frac{f}{B}-\frac{1}{2}\right)\right] \right.$$

$$\left. -j\, S\left[\sqrt{2TB}\left(\frac{f}{B}+\frac{1}{2}\right)\right] + j\, S\left[\sqrt{2TB}\left(\frac{f}{B}-\frac{1}{2}\right)\right] \right\}$$

where C(t) is the Fresnel integral of the first kind
S(t) is the Fresnel integral of the second kind.

It is also worthwhile to consider the second derivative of the phase associated with the waveform in both the time and frequency domains. In the limit of large time-bandwidth products, the group delay approximates the delay experienced by the instantaneous frequency (f_i) in the time domain. For example, the amount of energy in a compression type waveform in the frequency band (Δf) is given by $\frac{1}{2}|M(f)|^2\Delta f$. But this energy manifests itself in the time domain with a delay τ_g. Therefore

$$\frac{1}{2}\,|\psi(t)|^2 \Delta\tau_g = \frac{1}{2}\,|M(f)|^2 \Delta f$$

or

$$\frac{\Delta\tau_g}{\Delta f} = \frac{|M(f_i)|^2}{|\psi(\tau_g)|^2} = -\frac{1}{2\pi}\frac{d^2}{df^2}\,\theta(f)$$

where f is the instantaneous frequency of the signal at time (t).
For the linearly swept F. M. waveform, $\theta''(f)$ is a constant; hence, in the limit of large time-bandwidth product signals one expects the spectral amplitude to match the temporal signal envelope amplitude. Furthermore, if the linear F. M. waveform is amplitude weighted in either domain, the waveform in the opposite domain will mimic this same weighting.

Matched Filtering

Four central points which relate to the processing of large time-bandwidth (compression) pulses have been stated and derived above. They are:

1. There is a minimum time-bandwidth product for a pulsed signal.
2. For a given waveform amplitude distribution in the time domain, any non-linear phase function in time will increase the nominal and the rms bandwidth of the signal.
3. For a given waveform amplitude distribution in the frequency domain, any non-linear phase function in frequency will increase the nominal and the rms time duration of the signal.
4. For linear swept F. M. pulse compression signals (time-bandwidth products > 10) the signal envelope in one domain will mimic the signal envelope in the inverse (transform) domain. The signal phase will be quadratic in both domains as well.

Based on these rules, one way to achieve pulse compression in time is to run the signal through a filter which will remove the non-linear phase of the signal spectrum. This is accomplished by the so-called equi-phase network whose transfer function if given by:

$$H(f) = rect\left(\frac{f-f_0}{B}\right) e^{-j\theta(f)}$$

where θ (f) is the spectral phase of the compression waveform. If B (the nominal bandwidth of the filter) is sufficiently large, the non-linear phase of the spectrum of the compression waveform is removed and the output signal achieves the minimum time-bandwidth product. In so doing, the nominal signal duration of the input signal is greater than that of the output signal by the time-bandwidth product of the input signal. In as much as the nominal envelope of the "chirp" signal in the time and frequency domains are rectangle functions, the equi-phase network is a matched filter for the "chirp" signal. That is, the nominal transfer function of the equiphase network is the complex conjugate of the "chirp" signal spectrum. The compressed or "dechirped" wave form (the matched filter output) is a sinc function ($\sin \pi x / \pi x$). Thus, though most of the energy (better than 90%) contained in the original chirp pulse, is included in the main lobe of the compressed (in time) waveform, there are additional responses or side-lobes off the mainlobe. The peak sidelobe is 13.2 DB down from the peak main lobe of the response and this may be sufficient to cause serious difficulties in "reading" the output of a matched filter. The matched filtering which has been described is a linear, time invariant process in which the principle of superposition applies. Therefore, two targets, separated in time by the interval between the peak of the mainlobe and the peak of the principle side lobe of the matched filter response, with each target returning chirp signals to the filter may be interpreted as a single target with its attendant response sidelobes. The problem is substantially lessened by shaping the amplitude spectrum of the response.

Rihaczek[13] gives a particularly enlightening analysis of the general principle underlying spectrum shaping. Following Woodward,[14] he establishes a measure of the degree of ambiguity in the response of a matched filter. This number (A_τ), is equal to the normalized area under the squared envelope of the matched filter response. Richaczek shows that A_τ may also be described in terms of the normalized spectrum envelope ($M_n(f)$) of the input signal according to:

$$A_\tau = \int_{-\infty}^{\infty} |M_N(f)|^4 \, df$$

where $s(t) = a(t)\, e^{j\phi(t)}\, e^{j 2\pi f_c t}$

$$S(t) = M(f - f_c)$$

$$M_N(f) = M(f)/\sqrt{2E}$$

E is the energy of the compression waveform.

The ratio of that part of (A_τ) due to the main lobe of the filter response (A_o) to the total value of A_τ is reasonable approximated by:

$$\frac{A_z}{A_o} = 4\left[\int_{-\infty}^{\infty} f^2 |M_N(f)|^4 df\right]^{\frac{1}{2}} \int_{-\infty}^{\infty} |M_N(f)|^4 df$$

Thus in order to minimize the "ambiguity" of the filter response due to signal outside the main response lobe this ratio must be minimized. However, the second integral is directly proportional to the response signal energy which one chooses to keep constant. The first integral on the other hand, varies according to the rms bandwidth of the filter response. It can be controlled by signal spectrum shaping. Figure 7, taken from Rihaczek,shows two signal spectra. The matched filter response to the first of these two signals will have relatively high sidelobes while that for the second signal will have low sidelobes.

There is one obvious price that one pays for reducing the sidelobe levels in this manner. Indirectly, it stems from the fact that the equiphase network no longer approximates a matched filter. Recall that the transfer function of the equiphase network was flat in amplitude across its bandwidth. This "matched" the flat spectrum of the large time-bandwidth chirp signal across its spectrum and the equiphase network was a "matched" filter.

Consider a matched filter where the spectral amplitude of the filter response is the same as the spectral amplitude of the "shaped chirp signal". The output of the matched filter will have a reduced rms bandwidth if the signal spectrum is shaped for minimum sidelobes. Thus the rms duration of the compressed signal will not be as short as it would have been without shaping. What this amounts to, is that the response main lobe will be wider than without spectral shaping. Some close target resolution is sacrificed for the sake of reducing side lobe ambiguities.

If an equi-phase filter is employed to compress the linearly swept FM compression waveform, the ratio of the total ambiguity of the filter response to the ambiguity in the principal lobe of the filter response is given by:

$$\frac{A_z}{A_o} = 4\left[\int_{-\infty}^{\infty} f^2 |M_N(f)|^2 df\right]^{\frac{1}{2}} \int_{-\infty}^{\infty} |M_N(f)|^2 df$$

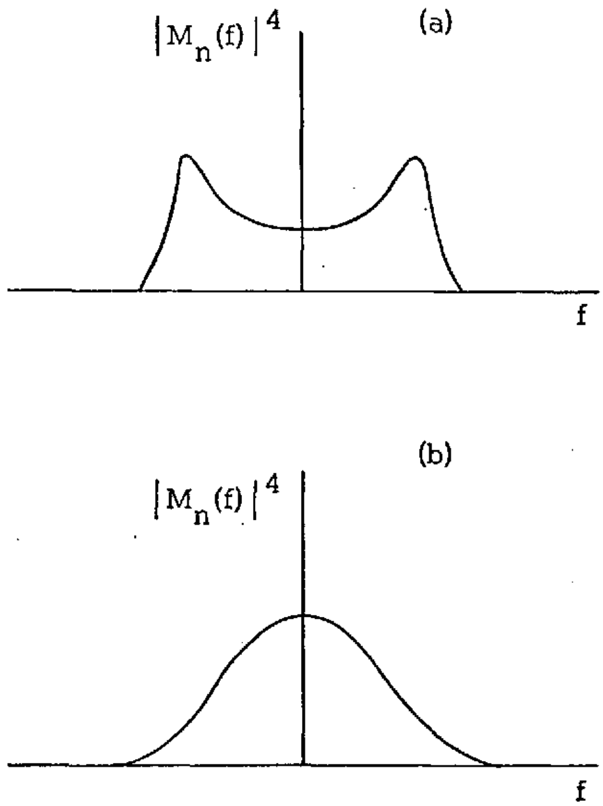


Figure 7. Spectra with a) high and b) low time sidelobes (From Reference 13, page 96).

Once again the sidelobe ambiguities are reduced as the rms bandwidth of the filter response decreases. It is interesting to note at this point that were it possible to tolerate high sidelobe ambiguities, a signal spectrum such as that shown in Figure 7 would achieve, on passage through a matched filter, a reduction in the duration of the main lobe of the response greater than the time bandwidth product of the input signal. This happens because the rms bandwidth of the output signal is greater than that of the input signal.

It was shown above that for linearly swept FM signals of large time-bandwidth products (greater than 10) the signal envelope closely mimics the signal spectrum envelope as a function of time and frequency, respectively. Because of this, filter transfer function amplitude shaping may be done in the time domain. Therefore, if it is desirable to realize equiphase filtering which has been modified by amplitude spectrum shaping, a convolution of the linearly swept F M compression signal with the shaped impulse response (in time) of the equiphase filter may be performed. As an example of the power of "envelope mimicking" in the time and frequency domains for linearly swept F M signals of large time-bandwidth products, consider a side-looking radar operating against a point target. Let the two-way voltage antenna pattern be a truncated gaussian function in the along track direction with the peak gain directed normal to the radar velocity vector. Assuming that it takes one second for the antenna beam to sweep past the target and that the illumination frequency is such that the total doppler bandwidth of the received signal is 100 Hz, the received signal (s(t)) from a point target will be:

$$s(t) = c_1 \, \mathrm{rect}(t) \, e^{-c_2 t^2} e^{j(2\pi f_c t - 100\pi t^2 + \phi)}$$

Then, according to the mimicking property

$$S(f) = c_1 \left(\tfrac{1}{10}\right) e^{-j\pi/4} \, \mathrm{rect}\left(\frac{f - f_c}{100}\right) e^{-c_2\left(\frac{f-f_c}{100}\right)^2} \times e^{j\left[\phi - 100\pi\left(\frac{f-f_c}{100}\right)^2\right]}$$

and the matched filter impulse response would be given by:

$$h(t) = s^{*}(-t) = c_1 \, rect(t) \, e^{-c_2 t^2} e^{-j(-2\pi f_c t - 100\pi t^2 + \phi)}$$

It should be noted that once the linearly swept F. M. signal has been processed by either a matched or an equiphase filter, the spectrum of the output signal is real. Therefore, the time domain response is the Fourier transform of the signal spectrum envelope in the case of an equiphase filter, or it is the Fourier transform of the signal spectrum envelope squared in the case of a matched filter. Such a relation also exists in the case of weighted antenna aperture illumination and its far field Fraunhofer antenna pattern. Numerous references exist, giving these transform pairs, and since the antenna designer is vitally concerned with reducing sidelobe strength, studies of optimum aperture weighting exist. These provide convenient references for filter spectral envelope and signal spectral envelope design. Thus, for example, it is known that a uniformly illuminated one dimensional antenna aperture produces a sinc function radiation pattern. Similarly then, a matched filter with a rectangular spectral envelope produces a compressed signal which is a sinc function. Sidelobe reduction due to Taylor spectral weighting has been well studied and documented.[9] There are in fact, many schemes documented which reduce sidelobes at a cost of decreased close target resolution (primary lobe width).

Quadrature Detection

To this point, the arbitrary phase factor (ϕ) in the signal has not been properly high-lighted. It has been assumed that this factor was matched by the filter. In reality, this factor is arbitrary, and cannot be matched in a given filter. However, its effect is masked when the matched filtering is done at an offset frequency with a passive filter in as much as envelope detection is applied to the filtered signal output. On the hand, if matched filtering is done by an active correlator, correlation with the reference function (against a length of stored continually updated received signal) at intervals shorter than the inverse of twice the signal bandwidth allows the modulating envelope of the output signal to be retrieved with low pass filtering again independently of ϕ. However, in as much as the effective doppler bandwidth from a point target is halved if the offset frequency is zero, the P R F of the radar system may be set equal to actual doppler bandwidth (with offset frequency the P R F is twice the doppler bandwidth). Because the radar system P R F is the determining factor in the width of swath which may be imaged, reducing the P R F by a factor 2 allows the swath to be doubled. Figure 8 displays a comparison of the continuous (c. w. illumination) waveform received by a side-looking radar from a point target; in the first instance on an offset frequency, and in the second with no offset frequency. For the case with no offset frequency, the effect of the arbitrary phase angle becomes visible.

Assume the reference function to be given by:

$$s_r(t) = rect\left(\frac{t}{T}\right) \cos k t^2$$

and the signal with no offset to be

$$s(t) = rect\left(\frac{t}{T}\right) \cos(k t^2 + \phi)$$

The correlation function is then given by:

$$R(\tau;\phi) = \int_{-\frac{T}{2}+\tau}^{T/2} \cos k t^2 \cos(k(t-\tau)^2 + \phi)\, dt$$

It is apparent that the peak value of R ($\tau ; \phi$) will occur with $\tau = 0$, but the value of R (0) will be a strong function of the arbitrary phase angle (ϕ). For example, if ϕ equals zero

$$R(0;0) = \int_{-T/2}^{T/2} \frac{dt}{2} + \int_{-T/2}^{T/2} \frac{\cos 2kt^2}{2}\, dt \approx \frac{1}{2}\left(T + \frac{1}{2}\sqrt{\frac{\pi}{k}}\right)$$

but if ϕ equals $-\pi/2$

$$R(0;-\pi/2) \approx \frac{1}{4}\sqrt{\frac{\pi}{k}}$$

and the ratio of R (0; 0) to R $(0;\frac{-\pi}{2})$ is ($1+2\sqrt{TB}$)

where (TB), the time-bandwidth product of the signal may be quite large.

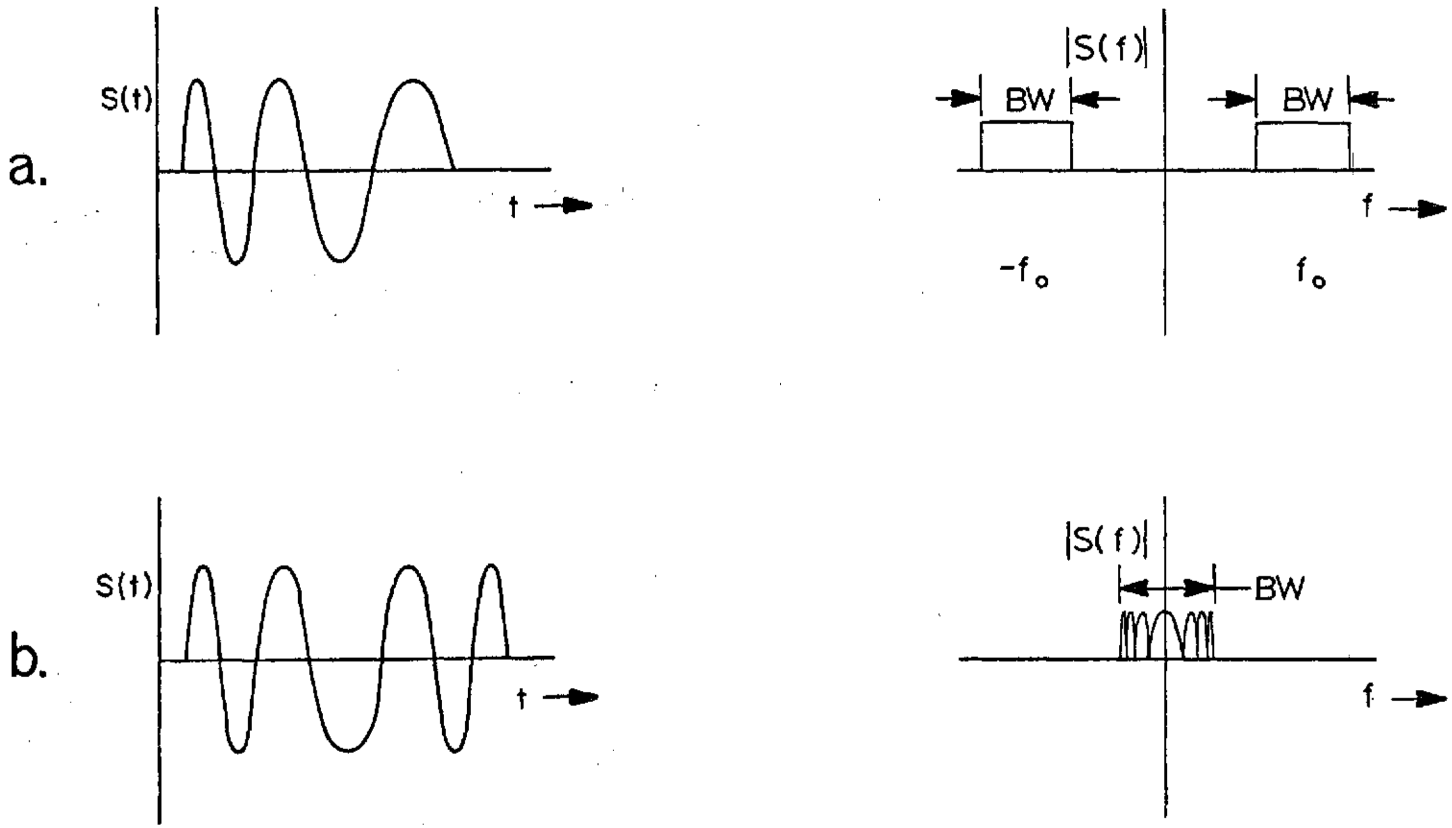


Figure 8. The same linearly swept F.M. signal (a) on an offset frequency, (b) with no offset frequency.

Algorithm Employed With Quadrature Detection

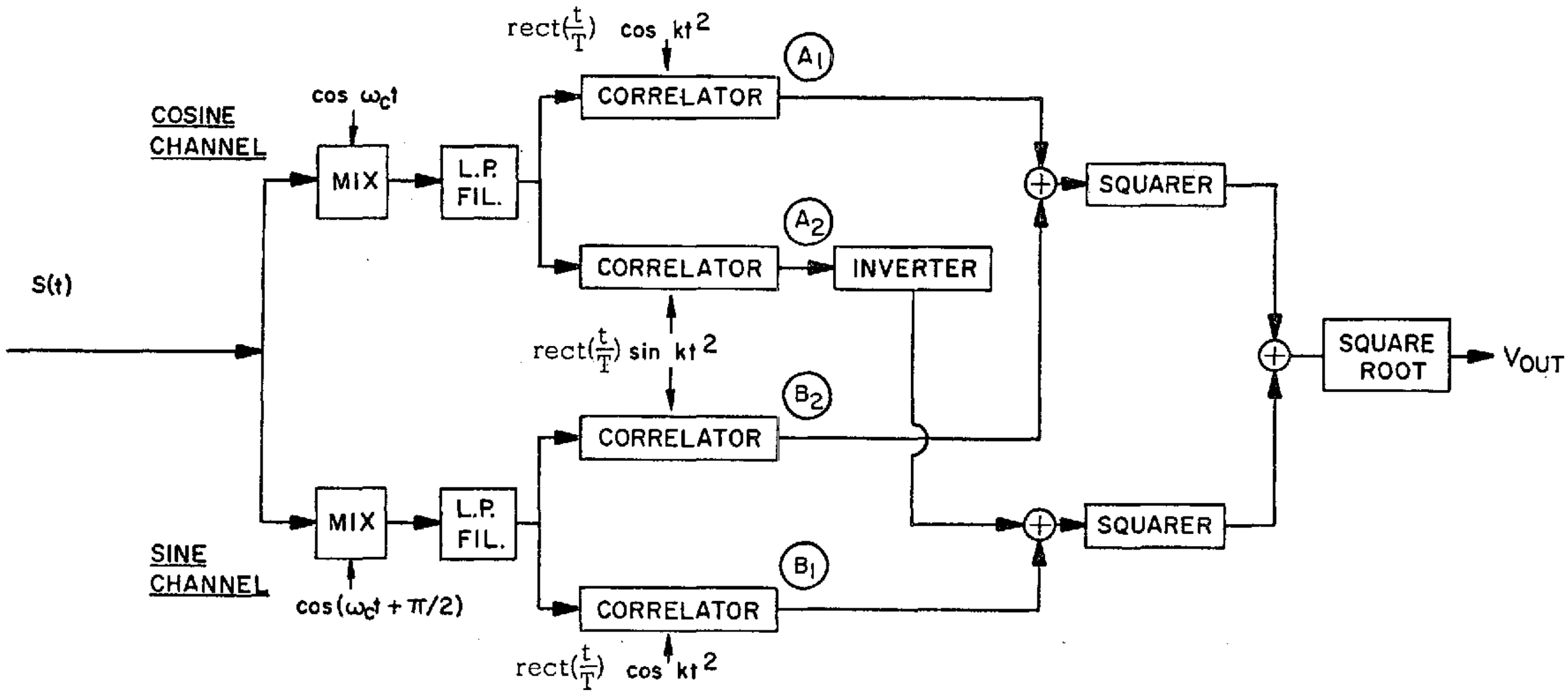


Figure 9.

Figure 9 is a schematic of the algorithm used when quadrature detection is employed in a side-looking radar. The input signal depicted is that due to a point target. The output of this algorithm is very nearly linear with input voltage amplitude and independent of (ϕ). According to Figure 9, the outputs of the four correlators are given by:

$$
\begin{array}{ll}
A_1 & A\,\overline{cc}\cos\phi - A\,\overline{sc}\sin\phi \\
A_2 & A\,\overline{cs}\cos\phi - A\,\overline{ss}\sin\phi \\
B_1 & A\,\overline{sc}\cos\phi + A\,\overline{cc}\sin\phi \\
B_2 & A\,\overline{ss}\cos\phi + A\,\overline{cs}\sin\phi
\end{array}
$$

where, for example,

$$\overline{sc} = \int_{-T/2+\tau}^{T/2} \sin kt^2 \cos k(t-\tau)^2\,dt$$

and following the rest of the algorithm, the output voltage is

$$V_{OUT} = A(\overline{cc} + \overline{ss})$$

or

$$V_{out} \approx 2A \int_{-T/2+\tau}^{T/2} \cos kt^2 \cos\left(k(t-\tau)^2\right) dt$$

Figures 28 and 28a in Chapter V show comparison of quadrature detection and non-quadrature detection for zero offset frequency. In these figures, a statistical target of given cross section is being imaged. Quantitatively the M/STD (mean image voltage to its standard direction) ratio in the final images is 5.5 dB and 2.5 dB for the quadrature and non-quadrature receiver respectively.

Side Looking Radar Ambiguity Diagrams

Ground mapping by synthetic aperture radars may be considered as a special case of general target detection radar. The targets for SAR are stationary and closely packed. The relative velocities of the targets and the radar are determined solely by the radar velocity and the direction in which its antenna is pointed. Both of these quantities are known. Thus, certain constraints exist on the parameters of relative motion of the targets with respect to the radar.

Implementation of radar return processing for synthetic aperture radar images is most often accomplished by cross correlating the stored returns with a particular reference function. To this point, synthetic aperture radar has been described in terms of matched or equi-phase filtering. However, filtering which is usually thought of as a frequency domain multiplication process, has its counterpart -cross correlation - in the time domain. There is little question but that linear passive filtering is the most simply effectual form of such processing. However, it suffers from the fact that a large number of returns collected over a time period of the order of hundreds of milli-seconds, must be processed at any instant. Passive filters with delay times of these magnitudes are impractical to build. Therefore, the processor requires a memory capability. With sufficiently fast memory readout , a passive filter could follow the memory and perform the dechirping or filtering function. Such a scheme may have merit, but it is not usual, and it will not be studied in this report. Instead, the string of radar returns stored in memory are multiplied by a reference function and the sum of products is taken for form a cross correlation.

Consider an illumination pulse train or the signal used to track a point target on the ground. The target has a radial position (R), a radial velosity ($\dot{R}$), and a radial acceleration ($\ddot{R}$). The effective pulse train duration operating against the point target is the time required to fly the along track beam width of the physical antenna at the target. The measure of the round trip time of the pulse train to and from the target will be taken in the time required for the center of the pulse train to make the round trip. Of course, the envelope of the return signal (pulse train) will be distorted to some extent due to the relative motion of target and radar over the pulse train duration. However, Rihaczek[13] has shown that envelope distortions due to relative velocity ($\dot{R}$) are inconsequential as long as the nominal time-bandwidth product of the pulse train satisfies the following inequality

$$TB \leq 0.1\, c/\dot{R}$$

where c is the velocity of light.

He has also shown that relative acceleration does not distort the envelope significantly, provided that

$$\tau^2 B \leq 0.2\, c/\ddot{R}$$

Both these conditions are usually satisfied in side-looking radar systems, where radial velocities and accelerations are small.

To establish a time reference, the time at which the pulse train center is launched is set equal to zero, and the time at which the center is received back is called τ_0. The delay or round trip time suffered by parts of the pulse train arriving back at the radar at time (t) may be expanded in a Taylor series according to the formula:

$$\tau = \tau_0 + \dot{\tau}_0(t - \tau_0) + \ddot{\tau}\,\frac{(t-\tau_0)^2}{2} + \text{orders of magnitude smaller terms}$$

where R_0 is the radial distance between target and radar at time $\tau_0/2$

$$\tau_0 = \frac{2R_0}{c}$$
$$\dot{\tau}_0 = \frac{2\dot{R}_0}{c}\left(1 + \frac{\dot{R}_0}{c}\right)^{-1} \approx \frac{2\dot{R}_0}{c}$$
$$\ddot{\tau}_0 = \frac{2\ddot{R}_0}{c}\left(1 + \frac{\dot{R}_0}{c}\right)^{-3} \approx \frac{2\ddot{R}_0}{c}$$

Since the return signal envelope is taken to be essentially the same as that of the illuminating signal, it is merely delayed by a time τ_0. However, the phase of the carrier will be sensitive to small differences in delay time over the duration of the return signal and this sensitivity will manifest itself in non-linear phase terms in the return signal.

Consider an illuminating pulse train of finite duration on a carrier frequency (f_0)

$$s(t) = a(t)\, e^{j2\pi f_0 t}$$

where $a(t) = \mathrm{rect}\left(\frac{t}{T_{PUL}}\right) * \left[\mathrm{comb}\left(\frac{t}{T_{PER}}\right) \cdot \mathrm{rect}\left(\frac{t}{T_{tr}}\right)\right]$

T_{PUL} = the basic pulse duration

T_{PER} = the pulse spacing (PRF^{-1})

T_{tr} = the length of the pulse train

The return signal will be approximated by

$$s_r(t) = a(t-\tilde{\tau}_0)\, e^{j2\pi f_0 (t-\tau)}$$

Letting the following substitutions be made,

$$\nu_0 = -f_0 2 \frac{\dot{R}_0}{c}$$

$$\gamma_0 = -f_0 2 \frac{\ddot{R}_0}{c}$$

the received signal may be rewritten as:

$$s_r(t) = a(t-\tau_0)\, e^{j2\pi\left(f_0(t-\tau_0)+\nu_0(t-\tau_0)+\gamma_0(t-\tau_0)^2\right)}$$

(ν_0) is recognized as the doppler frequency. Also, it is seen that the term due to target acceleration (γ_0) will contribute to a linear change in phase with time, creating a new effective doppler shift no longer associated with just the relative velocity of target and radar. In addition, the acceleration term (γ_0) will cause the return signal to be linearly swept in frequency with time.

To effect matched filtering, in the time domain, a cross-correlation is performed with the following reference function:

$$a(t)\, e^{-j2\pi\left(f_0 t + \nu_1 t + \gamma_1 t^2\right)}$$

The cross correlation is given by:

$$\chi(\tau, \nu, \gamma, \gamma_1) = \frac{e^{j2\pi(f_0+\nu_1)\tau}}{2} \int_{-\infty}^{\infty} a(x)\, a(x-\tau)\, e^{j2\pi(\nu x + \gamma x^2 + 2\gamma_1 x\tau - \gamma_1 \tau^2)}\, dx$$

where $x = t - \tau_0$

$$\nu = \nu_0 - \nu_1$$

$$\gamma = \gamma_0 - \gamma_1$$

The factor 1/2 comes from the fact that the signal is written in exponential form. Note here that τ, ν and γ are respectively the delay, doppler, and acceleration mismatches between the reference and received signals.

The usual form of ambiguity function defines a surface over the $\tau - \nu$ plane, whose height is the amplitude of $(\chi(\tau,\nu))$. In the usual case, higher range derivatives than the first are zero. One may plot the usual ambiguity diagrams for the case with range acceleration if γ and γ_1 are considered parameters. However, a simpler and more intuitively saitsfying procedure is to rewrite as follows:

$$\chi(\tau,\nu,\gamma,\gamma_1) = \frac{e^{j2\pi(f_0+\nu_1)\tau}}{2} \int_{-\infty}^{\infty} \left\{ e^{j2\pi(\gamma x^2 + 2\gamma_1 x\tau - \gamma_1 \tau^2)} \left[a(x)a(x-\tau)e^{j2\pi\nu x} \right] \right\} dx$$

Recognizing that χ is an inverse Fourier transform and applying the rule that multiplication in one domain in convolution in the other, one is able to write χ in the form:

$$\chi(\tau,\nu,\gamma,\gamma_1) = \frac{e^{j2\pi(f_0+\nu_1)\tau}}{2} e^{-j2\pi\gamma_1\tau^2} F_x^{-1}\left\{ e^{j2\pi(\gamma x^2 + 2\gamma_1 x\tau)} \right\} \underset{\nu}{*} M(-\nu) \underset{\nu}{*} \left(M(-\nu) e^{j2\pi\nu\tau} \right)$$

where M(f) is the Fourier transform of a (t). Had the target range been constant at R_0 and the target velocity been constant at $\dot{R}_0$ and had the target acceleration been zero over the pulse duration, χ would have been:

$$\chi(\tau,\nu) = \frac{e^{j2\pi(f_0+\nu_1)\tau}}{2} M(-\nu) \underset{\nu}{*} \left(M(-\nu) e^{j2\pi\nu\tau} \right)$$

and the ambiguity function would have been:

$$|\chi(\tau,\nu)| = \frac{1}{2} \left| M(-\nu) \underset{\nu}{*} M(-\nu)\, e^{j2\pi\nu\tau} \right|$$

Considering the function $\chi(\tau,\nu)$ for the non-accelerating case as $\chi_{\ddot{R}=0}(\nu)$ with parameter τ, one realizes that $\chi_{\ddot{R}\neq 0}(\nu)$ with parameter τ for the accelerating target is the output of a filter whose transfer function is exp[$j2\pi(\gamma^2x^2+2\gamma_1\tau x)$)] and whose input is $\chi_{\ddot{R}=0}(\nu,\tau)$.

$$\chi_{\ddot{R}=0}(\nu,\tau) \rightarrow \boxed{H(x)=e^{j2\pi\gamma^2x^2}} \rightarrow \boxed{H(x)=e^{j4\pi\gamma_1\tau x}} \rightarrow \chi_{\ddot{R}\neq 0}(\nu,\tau)$$

The first half of the filter (H (x) = $e^{j2\pi\gamma^2x^2}$) is a dispersive filter which broadens a wave-form of finite duration in ν . However, the amount of broadening is proportional to the mismatch in acceleration (γ). This mismatch is controlled by the along-track velocity of the radar which in the imaging radar situation is fixed. Hence, the first filter is not expected to have a major influence on the ambiguity diagram for the radially accelerating target. On the other hand, the second filter (H(x) = $e^{j4\pi\gamma_1\tau x}$), delays $\chi_{\ddot{R}=0}(\nu;\tau)$ uniformly in ν by an amount equal to $2\gamma_1\tau x$. The second filter is the essence of imaging radar. To see why this is so, one applies the formulas derived, to the finite pulse train of illumination. Recalling that the illumination envelope is given by:

$$a(t) = \mathrm{rect}\left(\frac{t}{T_{PUL}}\right) * \left[\mathrm{comb}\left(\frac{t}{T_{PER}}\right) \cdot \mathrm{rect}\left(\frac{t}{T_{tr}}\right)\right]$$

with inverse Fourier Transform

$$M(f) = T_{PUL}\,\mathrm{sinc}(T_{PUL} f) \cdot \left\{ T_{PER}\,\mathrm{comb}(T_{PER} f) * T_{tr}\,\mathrm{sinc}(T_{tr} f) \right\}$$

the ambiguity function for the non-accelerating target is roughly estimated to be that shown in Figure 10a. And roughly the ambiguity function for the accelerating function is skewed as shown in Figure 10b.

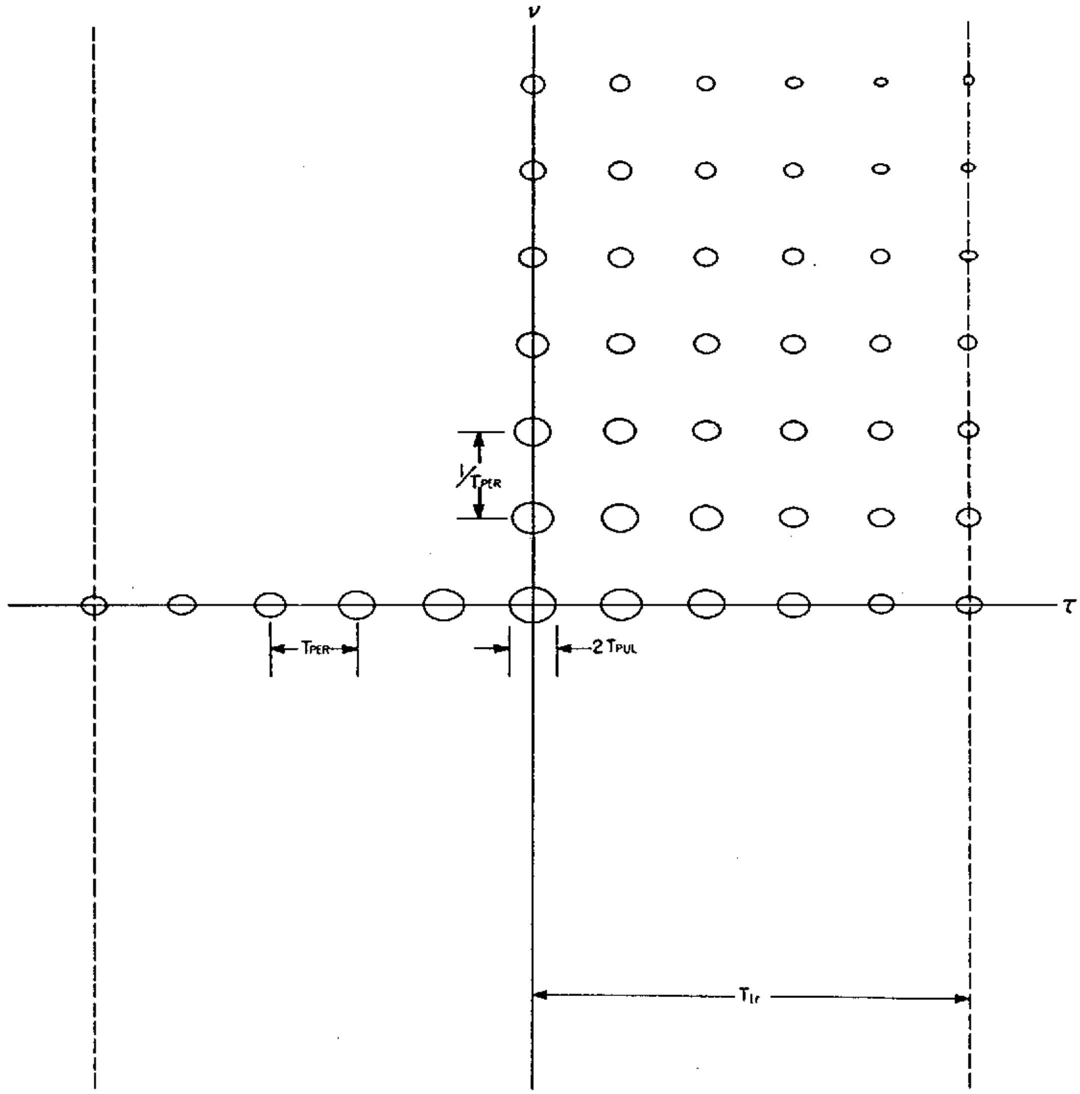


Figure 10a. Plan view of an ambiguity diagram for a target with constant range and velocity. T_{PUL} is pulse duration; T_{PER} is $(PRF)^{-1}$; T_{tr} is duration of illuminating pulse train.

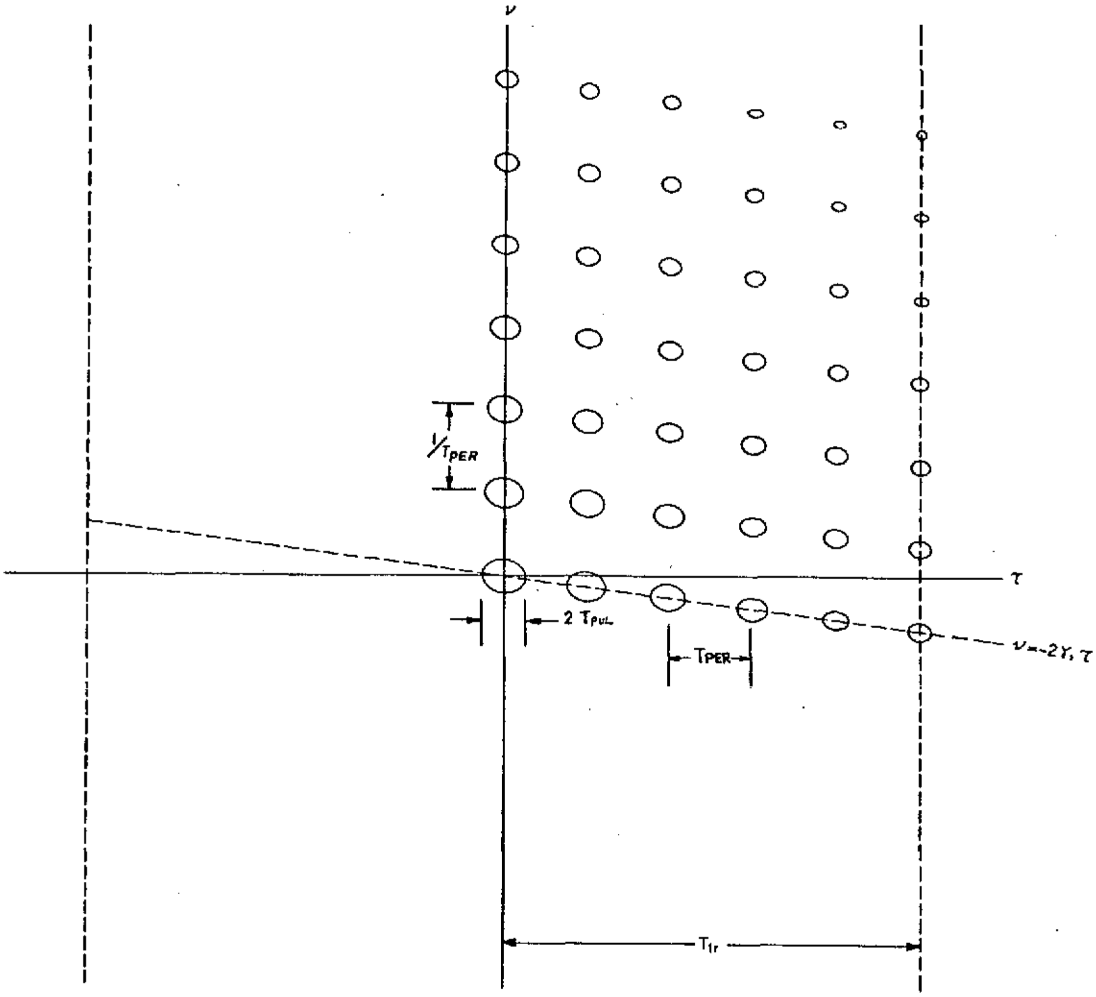


Figure 10b. Plan view of and ambiquity diagram for a target with slowly changing range and velocity and constant range acceleration ($\ddot{R}$) over the illuminating pulse train duration. T_{PUL} is pulse duration; T_{PER} is $(PRF)^{-1}$; T_{tr} is duration of illuminating pulse train; $\gamma_1 = \frac{2\ddot{R}}{c} f_o$.

Since the doppler mismatch is controlled by the speed of the radar platform as is the acceleration, the most important mismatch will be in delay. In order that there be no ambiguities on the delay axis, the diagram shows the following constraint

$$\frac{1}{T_{PER}} \geq 2\gamma_1 T_{tr}$$

For a side-looking imaging radar traveling at a velocity v_0 and with a uniformly illuminated antenna aperture ℓ meters long in the along-track direction

$$T_{tr} = \frac{\lambda}{\ell} \frac{R_0}{v_0}$$

and

$$\gamma_1 = f_0 2 \frac{\ddot{R}_0}{c} = \frac{2}{\lambda} \frac{v_0^2}{R_0}$$

yielding

$$\frac{1}{T_{PER}} \geq 2(\mathrm{BW})\ \text{doppler}$$

the Nyquist sampling criterion. The resolution in doppler is $1/T_{tr}$ corresponding to an along-track length of $\ell/2$. There are no ambiguities due to Doppler mismatch, since the separation between ambiguities is better than twice the doppler bandwidth possible with the geometry of synthetic aperture imaging. From the ambiguity diagram it is clear that the constant radial acceleration of the target during the illuminating pulse train is required to achieve fine along-track resolution. The acceleration skewes the ambiguity diagram removing delay ambiguities when the doppler and the acceleration terms are perfectly matched by the reference function.

VARIATIONS ON SYNTHETIC APERTURE RADAR PRINCIPLES

Synopsis

Variations on synthetic aperture radar imaging procedures are reflected in the modification of the processing algorithm introduced in Chapter II and sketched in Figure 9. The algorithm is studied in the frequency domain and its basic mechanism is reviewed. In this way the difference between quadrature and non-quadrature processing is established. Essentially the clutter level goes up and the mean to standard deviation ratio in the image of a Rayleigh distributed target decreases by 3 db with non-quadrature processing. However, the along-track resolution is the same for both types of processing. Non-quadrature processing eliminates 75% of the computations required in quadrature processing and cuts processor storage by half.

Zone Plate processing, presumming, and the effects of system and quantization noise are discussed. The notion of Subaperture processing is put forward as a method of reducing processor storage while simultaneously degrading image resolution to the required level and enhancing image mean to standard deviation (M/STD) ratio. The method is shown applicable in both quadrature and non-quadrature processing.

Introduction

All side looking image radar systems utilize the basic imaging geometry of target and radar described in Chapter I and illustrated in Figure 1. Hence in all systems, the same target histories would be available for filtering. However, the types of processing (filtering) that are possible are legion in number. Each offers a trade in the suitability of its final image to satisfy user requirements against the engineering requirements of an SAR system. An obvious example of this is the fact that matched filtering, or employing a reference cross correlation function which is the return signal history of a point target, provides the best close target resolution possible in the along-track direction. However, such processing is distorted to a degree which may or may not be acceptable, by the presence of minor but finite responses of the radar, to targets outside the principal beam of the synthetic aperture radar but within the beam of the radar's physical antenna. In Figure 16 one observes that a point target at a given range is imaged as a strong target of small but finite along-track width in the presence of minor targets at the same range. In Figure 17 the same point target is imaged as a strong target of slightly greater along-track width in the presence of much weaker targets at the same range. A trade has been made between close target resolvability and "clutter" or spurious response suppression. The trade was effected by using a weighted reference function in the second case. In so doing, though it is not apparent in Figures 16 and 17, the image-signal-to-system-noise decreased. However, since the clutter response of the system is a kind of noise the net gain (plus or minus) in image quality requires that this be taken into account.

Usually the primary requisite of a side-looking synthetic aperture radar system is to enhance the along-track close-target resolution in the image beyond that which is attainable by the radar's physical beam. When it is not necessary to achieve the best close-target resolution possible trades may be made leading to the design of simpler processors which just achieve the design requirements.

Beside the noise introduced into the synthetic aperture radar image due to system electronics and clutter ambiguities noise is introduced into the picture by the subject of the radar image itself. That is, the nature of extended terrain targets is such that their voltage reflection coefficients are statistically distributed. Two distinct plots of wheat, at the same slant range, will not return the same strength radar signal nor will the phase of the return carrier be the same. However, it is generally accepted that the mean value of the strength of the signal return is characteristic of wheat and different from that of other targets. Ideally one would like to obtain a map of target categories rather than an image of actual radar returns. The variance in the intensity of radar return strength from a given target category represents uncertainty for the user's purposes just as does system or clutter noise. With this in mind it becomes possible to improve image quality under certain conditions even though signal-to-system-noise decreases.

To this point, only fully-focused synthetic aperture systems have been discussed. They involve correlating a string of return signals with the impulses response of an equiphase filter whose transfer function amplitude may be weighted or with that of a matched filter. The nominal bandwidth of these filters has been equal to the doppler bandwidth of the returned signal (i.e., the nominal bandwidth of the impulse response, derived from an impulsive target at a given slant range over which the antennas physical beam is swept). According to the sampling theorem, this requires that a minimum number of pulse returns, equal to twice the time-bandwidth product of the return from a point target, be stored in the processor per range bin for processing at any instant.

The storage required in the processor is staggeringly large! This high storage capacity is the reason why film is the storage medium most often used. However, use of film precludes real time processing. The essence of the problem of real-time synthetic aperture image generation is the amount of processor storage required. It will be shown in Chapter VI that, although the number of electronic operations on the stored returns in the processor is also very large, synthetic aperture processing may be carried out electronically provided adequate storage is feasible.

All forms of processing studied in what follows involve some modification of the processing algorithm sketched in Figure 9 on page 32.

Non-Quadrature Detection

Costs involved in employing non-quadrature detection are discussed here. Without quadrature detection, all the algorithm sketched in Figure 9 page 32 are done away with save for A_1. Obviously there is a considerable saving in the number of arithmetic operations that are required for an image. Nearly 75 per cent of the operations are eliminated, the hardware required to implement the algorithm is reduced almost by a factor of 2, and most important of all, the storage required in the processor is halved. This is a huge savings.

The cost of this saving is most easily realized with the aid of the processor algorithm development of Chapter II and through Figures 30 through 30a on pages 128 and 129. The algorithm development shows that the image for a point target with only correlator (A_1) being used is:

$$A \Big/ \cos\phi \int_{-\frac{T}{2}+\tau}^{\frac{T}{2}} \cos kt^2 \cos k(t-\tau)^2\, dt$$

$$- \sin\phi \int_{-T/2+\tau}^{T/2} \sin kt^2 \cos k(t-\tau)^2\, dt \Big/$$

Table I shows the spectra for the various wave forms involved in the integrals.

The first integral in the point image expression is the auto-correlation of

$$\text{rect}\, \frac{t-\frac{T}{2}}{T} \cos kt^2 + \text{rect}\, \frac{t+\frac{T}{2}}{T} \cos\left(-kt^2\right)$$

The energy contained in each of these two terms is 1/4 that in the principal lobe response and it is spread uniformly in a low pedestal to one side or the other of the principal lobe over a time 2T. This is shown clearly in Figure 30 on page 128 (bottom curve) which is essentially the output of matched filtering of a V-FM waveform.

The second integral term in the output of correlator (A_1) for a point target image is the cross correlation of

$$\mathrm{rect}\,\frac{t-\frac{T}{2}}{T}\cos kt^2 + \mathrm{rect}\,\frac{t+\frac{T}{2}}{T}\cos(-kt^2)$$

with

$$\mathrm{rect}\,\frac{t-\frac{T}{2}}{T}\sin kt^2 - \mathrm{rect}\,\frac{t+\frac{T}{2}}{T}\sin(-kt^2)$$

In the frequency domain the following multiplication takes place

$$\frac{1}{2}\left[e^{j\frac{\pi}{4}}e^{-j\frac{\pi^2}{k}\nu^2} + e^{-j\frac{\pi}{4}}e^{j\frac{\pi^2}{k}\nu^2}\right]\cdot\left[e^{-j\frac{\pi}{4}}e^{-j\frac{\pi^2}{k}\nu^2} + e^{j\frac{\pi}{4}}e^{j\frac{\pi^2}{k}\nu^2}\right]$$
$$\cdot\left[T\sqrt{\frac{\pi}{k}}\,\mathrm{rect}\left(\frac{\nu-\frac{k}{\pi}\frac{T}{2}}{\frac{k}{\pi}T}\right)\right]^2$$

This time only the terms responsible for the pedestals remain in the product and the principal lobe of the response is gone. In fact the last curve of Figure 30a on page 129 is essentially the cross correlation of $\sin kt^2$ and $\cos kt^2$ and it shows only the pedestal.

Therefore, using non-quadrature processing, one expects an image with a reasonably large amount of clutter. Also the presence or absence of image points will depend on the phase of target voltage reflection

TABLE I

Positive Frequency Spectra of Up and Down Chirped Signals

$$f(t) = \mathrm{rect}\,\frac{t-\alpha}{T}\cos kt^2 \quad \text{for } \alpha \geq \frac{T}{2}\ ;\ k \text{ positive}$$

Exponential form	Positive freq. spectrum
$f(t) = \mathrm{rect}\,\frac{t-\alpha}{T}\, e^{jkt^2}$	$S(f) = \mathrm{rect}\left(\frac{f-\frac{k}{\pi}\alpha}{\frac{k}{\pi}T}\right) T\sqrt{\frac{\pi}{k}}\, e^{j\frac{\pi}{4}}\, e^{-j\frac{\pi^2}{k}f^2}$

$$f(t) = \mathrm{rect}\,\frac{t+\alpha}{T}\cos(-kt^2) \quad \text{for } \alpha \geq \frac{T}{2}\ ;\ k \text{ positive}$$

$$f(t) = \mathrm{rect}\,\frac{t+\alpha}{T}\, e^{-jkt^2} \qquad S(f) = \mathrm{rect}\left(\frac{f-\frac{k}{\pi}\alpha}{\frac{k}{\pi}T}\right) T\sqrt{\frac{\pi}{k}}\, e^{-j\frac{\pi}{4}}\, e^{j\frac{\pi^2}{k}f^2}$$

$$f(t) = \mathrm{rect}\,\frac{t-\alpha}{T}\sin kt^2 \quad \text{for } \alpha \geq \frac{T}{2}\ ;\ k \text{ positive}$$

$$f(t) = \mathrm{rect}\,\frac{t-\alpha}{T}\, e^{jkt^2} e^{-j\frac{\pi}{2}} \qquad S(f) = \mathrm{rect}\left(\frac{f-\frac{k}{\pi}\alpha}{\frac{k}{\pi}T}\right) T\sqrt{\frac{\pi}{k}}\, e^{-j\frac{\pi}{4}}\, e^{-j\frac{\pi^2}{k}f^2}$$

$$f(t) = -\mathrm{rect}\left(\frac{t+\alpha}{T}\right)\cdot\sin(-kt^2) \quad \text{for } \alpha \geq \frac{T}{2}\ ;\ k \text{ positive}$$

$$f(t) = \mathrm{rect}\,\frac{t+\alpha}{T}\, e^{-jkt^2} e^{j\frac{\pi}{2}} \qquad S(f) = \mathrm{rect}\left(\frac{f-\frac{k}{\pi}\alpha}{\frac{k}{\pi}T}\right) T\sqrt{\frac{\pi}{k}}\, e^{j\frac{\pi}{4}}\, e^{j\frac{\pi^2}{k}f^2}$$

In the frequency domain, the following multiplication takes place

$$\frac{1}{2}\left[T\sqrt{\frac{\pi}{k}}\;\mathrm{rect}\left(\frac{f-\frac{k}{\pi}\frac{T}{2}}{\frac{k}{\pi}T}\right)\right]^2 \cdot \left[e^{j\frac{\pi}{4}}e^{-j\frac{\pi^2}{k}f^2}+e^{-j\frac{\pi}{4}}e^{j\frac{\pi^2}{k}f^2}\right]$$

$$\cdot\left[e^{-j\frac{\pi}{4}}e^{j\frac{\pi^2}{k}f^2}+e^{j\frac{\pi}{4}}e^{-j\frac{\pi^2}{k}f^2}\right]$$

Which is equal to

$$\left[T\sqrt{\frac{\pi}{k}}\;\mathrm{rect}\left(\frac{f-\frac{k}{\pi}\frac{T}{2}}{\frac{k}{\pi}T}\right)\right]^2\left(1+\frac{1}{2}e^{-j\frac{\pi}{2}}e^{j\frac{2\pi^2}{k}f^2}+\frac{1}{2}e^{j\frac{\pi}{2}}e^{-j\frac{\pi^2}{k}f^2}\right)$$

In the above expression, the rectangle function goes to a sinc function in the time domain yielding the principal lobe of the response. Both of the remaining terms contain quadratic phase factors and aside from the constant phase factor, they may each be associated with a chirp function in the time domain through the transforms listed in Table I. (The constant phase term only accounts for a slight delay on the time axis.) The second term in the parenthesis transforms to:

$$\mathrm{rect}\left(\frac{t+T}{2T}\right)\cos\left(\frac{-kt^2}{2}\right)$$

The last term in the parenthesis transforms to:

$$\mathrm{rect}\left(\frac{t-T}{2T}\right)\cos\frac{kt^2}{2}$$

coefficients. Therefore, one might easily be led to expect that the mean to standard deviation ratio for images of random phase targets under non-quadrature processing would decrease. That is to say that the images of such targets would appear more granular than they would under quadrature processing. In fact the data that were gathered under this type of processing for targets with statistically distributed reflection coefficient phases indicate that the mean to standard deviation ratio of the images suffers a 3 db loss under non-quadrature processing vis a vis quadrature processing (see Table IV page 93).

It is interesting to observe the spectrum of a point target under quadrature detection. According to Chapter II the image is proportional to the absolute value of the sum of the auto-correlations of $\sin kt^2$ and $\cos kt^2$. Going back to Table I page 47 and computing the products of transforms shows that the pedestal or clutter response of each correlation exactly cancels out that of the other and that the principal lobe of each correlation reinforces that of the other. Therefore, close-target resolution will be the same independent of whether quadrature or non-quadrature processing is used. Studies 13 and 14 on pages 89 and 90 indicate that this is in fact the case.

Focused and Zone-Plate Processing

A saving in the processor complexity may be effected by modifying the four reference functions with which the input signal is correlated in quadrature processing, so that $\sin kt^2/|\sin kt^2|$ and $\cos kt^2/|\cos kt^2|$ is substituted for $\sin kt^2$ and $\cos kt^2$ respectively. Studies 3 and 7 on page 85 and 86 provide a comparison of the synthetic aperture beams employing the unmodified and the modified reference functions respectively. Except for slightly greater clutter levels in the case of the modified reference functions, there is not much difference. The modified reference functions are easy to implement in a digital processor since the values of the reference function are either 1 or -1. The modified reference functions are Fresnel Zone plates and hence the processing may be descriptively termed "Zone Plate" processing. Unfocused synthetic aperture radar is "Zone Plate" processing employing reference functions which include only the first Fresnel Zone. This kind of processing provides along-track resolution which is considerably poorer than Full Zone Plate processing as is shown in Figure 21 page 111. The TB products of the reference functions equal 2 in unfocused processing. And since T increases as the square root of slant range (R), the along-track resolution which is $v_o(B^{-1})$ (where v_o is the along-track radar velocity) is also proportional to the square root of the slant range. This is in contrast to full Focused or full Zone Plate processing where TB is directly proportional to R as is T and the along-track resolution is therefore also independent of R.

Presumming

Presumming is a technique commonly employed to reduce the storage required in a processor. It involves summing a series of return signals prior to processing them. Hence the storage required in the processor is down by a factor equal to the number of pulses summed. Of course to match this, the lengths of the reference function are also "compressed" in the same way. The effect of presumming is most easily realized by thinking of the process as a convolution of both the returns and reference functions in a given range bin with a rectangle function whose width is N(PRF) where N is the number of returns summed. Convolution in the time domain is the counterpart of multiplication in frequency domain and hence it becomes clear that presumming N pulses in the time domain is the same as reducing the doppler bandwidth of the return signal by N. But the along-track resolution of the processor is directly proportional to the inverse of the Doppler bandwidth and therefore the along-track resolution must degrade in direct proportion to the number of pulses summed. This is essentially the result shown for Study 10 pages 101 and 106 on presumming.

Image Quality

A well accepted model of terrain as a radar target is a grouping of point scatterers whose voltage reflection coefficient amplitudes are Rayleigh distributed and whose voltage reflection coefficient phases are uniformly distributed between π and $-\pi$ (see Appendix I). Targets which comprise a single terrain category are assumed to have a mean amplitude of reflection coefficient which is proportional to the square root of the differential scattering cross section of the target category. The radar with the best possible resolution attempts to create a map which is proportional to the amplitude of the individual point target reflection coefficient. For a given target category the Rayleigh distribution provides a mean to standard deviation in the image voltage which is $\sqrt{3.5}$. This means that there is a strong "speckling" in the image of a single target category. However, if the resolution is fine enough, the viewer of such a speckled image, may view it from a sufficient distance that a degree of image resolution is lost since the resolving ability of the eye decreases with distance. At the same time, the eye may be thought of as performing a convolution of the image with a two-dimensional "spot" whose diameter is greater than the image resolution length. This post detection integration will increase the M/STD ratio of the image to the viewer by the square root of the number of resolution cells in the "spot". The viewer has been enabled by this means to exchange image resolution length for gray-tone resolution. In many applications this viewer option is worthless inasmuch as the acceptable gray-tone resolution and length resolution are decided a priori on the basis of the type of terrain to be imaged. However, the size of the storage in the SAR processor and the bandwidth requirement of the telemetry link to transmit the image to the ground are vitally affected by the image resolution length. Therefore, what one seeks is some method of degrading image resolution in the processor to that required, and at the same time enhancing image quality through post-detection integration.

In SAR systems the image M/STD ratio is also decreased by system noise, clutter and in digital processing by quantization noise. However, both the effects of system and quantization noise are reduced to a remarkable degree by the large amount of predetection integration which takes place in the processor. The process of noise reduction in "matched filters" which is the essence of SAR radar processing is well documented in the literature.

In digital processing quantization noise is added to the returns which are to be match filtered. If the quantization increment is δ, Schwartz[8] shows that the mean error in each return, assuming returns which are distributed over a range which is large compared with δ is zero and that the variance of the error is $\delta^2/12$. Some insight into the magnitude of the error caused by quantizing returns for Full Focused quadrature processing may be had through the following computation. Assume that the synthetic aperture is (N) pulses long (2N numbers per range bin). The quantization increment is (δ). The rms error will be taken to be ($\delta/\sqrt{N\,12}$). The amplitude of return signal voltage from a resolution cell (A) will yield an image voltage (NA). Therefore the effective quantization noise is ($\delta^2/12N$). In Figure 22 page 112, A ranged between zero and 20. δ was 50 (5 bit quantization over a range of voltage from -800 to +800 in return signal) and N was 120. Therefore, the effective noise voltage was 50/46 or about 1 volt. The image is fairly good when S/N is better than 6 db and the image is very good beyond the first fifth of the image line. (see plot 1 Figure 22c, page 113). For 4 bit quantization the image should be relatively good after the first 40% of the image line and for 3 bits after the first 80% of the image line. These expectations are fairly well born out by the data in Figure 22c. In general, the maximum size of the quantization level which is tolerable will depend on the number of returns which are predetection integrated and on the range of image voltage expected.

Subaperture Processing

Subaperture processing is a method of improving the image mean to standard deviation ratio at expense of image along-track resolution. It offers as well the possibility of reducing the amount of storage required in the processor. Essentially the method creates several synthetic aperture antennas, each with a wider along-track synthetic antenna beam than is possible with full focused processing but still much reduced in along-track width compared to the physical antenna's beam. Each of these beams is "squinted" within the width of the physical antenna's beam so that it is directed fore or aft of broadside to the radar's track. It turns out that images achieved with subapertures whose principal lobes do not embrace the same range of doppler frequencies in the return signal are images of ground targets from a given statistical population which are decorrelated. Hence such images, once detected, may be summed to increase the mean to standard deviation ratio of the final image. Essentially, one is able to perform averaging in Doppler bandwidth to reduce image graininess. This is in contrast to the theses of W. P. Waite[16] and G. Thomann[17] who have studied averaging in range bandwidths to reduce image variance. One special advantage which accrues to averaging in doppler bandwidth is that it allows one to employ synthetic aperture processing techniques to reduce along-track resolution. So far this has not been shown to be possible with averaging in range bandwidth since the coherence required for synthetic aperture radar is absent. That is, the averaging is done post-detection.

Subaperture processing is accomplished by using only a portion of the full length of the reference functions which are used in full focused processing. Thus in full focused processing, correlations are performed against reference functions of the form

$$\mathrm{rect}\,\frac{t}{T}\ \begin{matrix}\cos\\ \sin\end{matrix}\left\{kt^2\right\}$$

For subaperture processing the reference functions go to

$$\mathrm{rect}\ \frac{t \pm \alpha}{x}\ \begin{matrix}\sin\\ \cos\end{matrix}\left\{ k t^2 \right\}$$

Here $|\alpha| \leq \left(\frac{T-x}{2}\right)$ and $x \leq T$. If one desires 5 non-overlapping subaperture beams, the total or complete correlation reference function could be divided into 5 non-overlapping parts. The number of returns stored in a range bin at any instant would be 1/5 that required for full focusing. However, the number of correlations required would be 5 times greater than would have to be done on each set of numbers in a range bin under full focussed processing because essentially 5 different images of a range line would be in the process of being generated at any instant. Nevertheless, there would still be an increase in the time available for computation of the order of 5 because decreasing the resolution by a factor of 5 means that the image line being generated need only be sampled one fifth as often. Hence by using such subaperture processing only 1/5 the processor storage is required and there is 5 times as much time to make computations as there is for full focused processing and the image quality vis a vis the M/STD ratio is up by a factor of 5 in the final image. The price is that the along-track resolution has been degraded by a factor of 5. The impulse response of such non-overlapping subapertures is shown in Figures 26 and 27 on pages 120 and 122.

The one element of increased processor storage not mentioned is the temporary storage required for each of the 5 imaged lines generated per range bin. That is, since the five synthetic aperture beams are non-overlapping, the images cannot be summed completely until the radar has moved a distance equal to (N-1)/N times the length of the physical antenna's beam on the ground. This storage is not inconsiderable. However, it is extremely slow storage and might easily be done outside the primary processor with a storage technology much less sophisticated that that required to perform the correlations. It would not be unusual for the cycle times required in the "averaging stores" to be three orders of magnitude greater than that required of the correlation or "working stores."

Gerchberg and Haralick[18] have shown that depending on the covariance of samples from a statistical population the M/STD may be increased by an optimum weighting of correlated samples. If this technique proved practicable in the subaperture processing situation it might be possible to either increase the M/STD ratio in the image by a factor greater than the along-track image degradation or to reduce the total number of images and hence the size of the averaging stores while retaining a given M/STD ratio.

The technique of subaperture processing applies to non-quadrature processing as well. Of course, using non-quadrature processing reduces the size of the "working stores" by a factor of two and the number of computations by about 75 per cent. At the same time the mean to standard deviation (M/STD) ratio decreases by the factor $\sqrt{2}$. This is documented for Rayleigh distributed targets in Table IV page 93.

It is instructive to study the point target response for non-quadrature subaperture processing in Figures 27 and 27a on pages 122 and 123. Of course one must be aware of the fact in studying these Figures that they are for a point target with a particular reflection coefficient phase. They are dependent, especially in regard to the size of the principal response lobe, on the value of this phase (Study 18 page 98). The central subaperture response (curve 3 of Figure 27a) is essentially the same as one would expect from matched filtering of V-FM as discussed earlier in this chapter. However, the other four responses shown in Figure 27a have the pedestal separated from the primary lobe. By using the Fourier Transforms shown in Table I one can show that the pedestals arise from the cross correlation of an increasing frequency chirp with a decreasing frequency chirp over the same band width. Since the reference function for these subapertures (all except the middle subaperture whose reference function is a V-FM signal) are either increasing or decreasing linearly swept FM signals, the primary lobe will be created when the reference function lines up with that part of the V-FM point target return signal which has the same spectrum as the reference function does to within a complex constant.

However, the pedestal is created as that part of the point target signal occupying the same bandwidth as the reference function, albeit chirped in the opposite sense to the reference function, is cross correlated with the reference function. Thus in the topmost plot of Figure 27a the pedestal and the main lobe are separated essentially by the along-track width of the physical antenna's beam on the ground. This is because the reference function which creates this subaperture corresponds to the bandwidth of the point target signal as the first fifth of the physical beam sweeps by the point. Of course for the V-FM generated by the point, the same band is again encountered (although the chirp is in the opposite sense) as the last fifth of the physical beam sweeps by the point.

Figure 29a of Study 17 shows a comparison of subaperture imaging with quadrature and non-quadrature processing. These pictures give some insight into the meaning of M/STD ratio which has been adopted in this paper as an image figure of merit.

PROCESSOR SIMULATION

Synopsis

A Fortran program was written to have a large general purpose digital computer simulate the imaging performance of various radar systems against diverse targets. The program is documented in Appendix III. The program itself contitutes a major product of the research performed. Therefore, this chapter is written to link the program parameters to the physical phenomena which they model. It also elucidates the numerous processing options which are at the user's disposal.

Introduction

A major problem in creating the computer program to simulate a digital processor was that of making the program sufficiently flexible. Processors may be more or less effective depending on the nature of the targets imaged. Performance also depends on the linearity of the receiver characteristic, the receiver noise figure, possible Doppler mismatch, the antenna pattern and the degree of quantization employed in the digital processor. For a full evaluation, these characteristics had to be included in the simulation program along with a large variety of possible processing techniques. Provision made for these features is detailed below.

Another problem was to devise some effective method of evaluating processor performance. Three separate devices are used for this purpose in the program:

1. A plot routine built into the program yields a comparison of the actual along-track targets in a given range bin with the image line generated by the processor. In effect this plot routine graphs the

amplitude of target reflection coefficients in a line and it also graphs the image of these amplitudes.

2. Statistical analyses of target reflection coefficient amplitudes, radar return signals and the image line generated by the processor are made. Means, variances and covariances are determine as well as 90 per cent confidence interval for these results. The statistical analysis is particularly important in determining the extent to which the sub-aperture technique described in Chapter III eliminates image graininess.

3. A full 2-dimensional picture can be generated. This method can be deceptive; however, used with care it does present a satifying example that makes quantitative statistical results more meaningful.

The simulation program is documented in detail in Appendix III. There, the reader will find the overall program organization as well as the organization of each subroutine which comprises the program. A printout of the program is also included as part of the Appendix. In what follows here, the user's options and input methods for them are described. The options are listed in terms of the physical phenemena they model. Ten cards are input to the program per run. A list of subheadings in the remainder of this chapter is presented below:

A. Ground target type
B. Physical antenna pattern along track
C. Target line pattern options
D. Pulse repetition frequency
E. Receiver gain function
F. Receiver noise
G. Doppler mismatch
H. Signal quantization
I. Focused and zone plate processing options
J. Program output options
K. Statistical analysis of processor performance

Figure 11 shows a schematic diagram of the radar system components and organization which is being modeled.

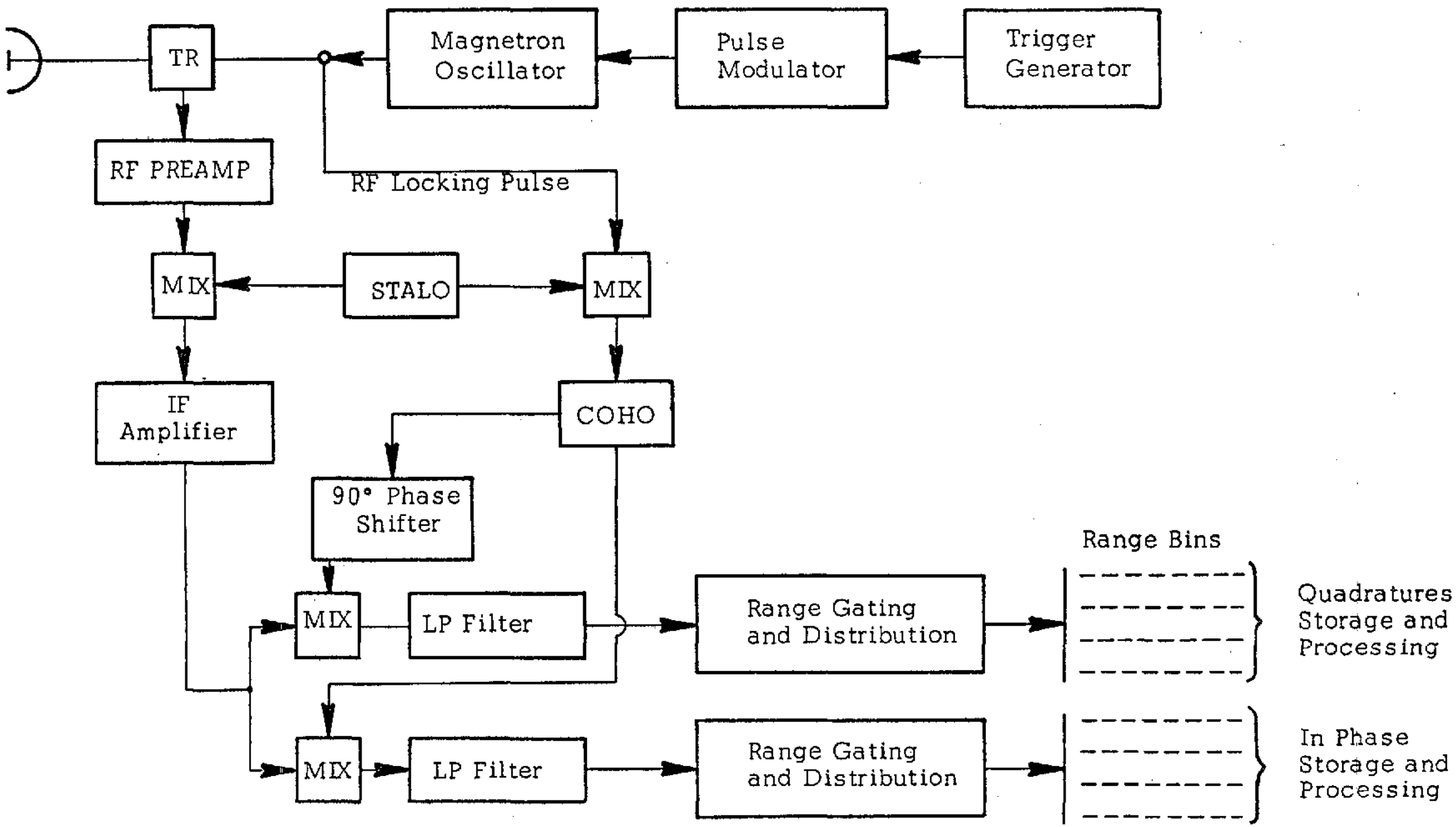


Figure 11. Schematic of SAR System Simulated by Computer Program.

The simulation program is organized to image only a single line of targets (one range). The number of such bins which must be processed in parallel in an actual processor is found by dividing the slant range swath (cross-track) to be imaged by the cross-track slant range resolution length.

Ground Target Type

The slant range resolution cell is taken to consist of point targets, equally spaced along the line. In specifying the type of targets which comprise the image line, the computer requires numbers proportional to the mean voltage reflection coefficient of the target and to the phase of the coefficient as well. Provision has been made to input arbitrary target lines by completing the empty subroutine EXTARG (Appendix III page 249) or the user may use internally generated targets which fall into four distinct classes:

1. The target reflection coefficient amplitudes are Rayleigh-distributed with a mean equal to 1.25 times a user supplied constant and a variance equal to .43 times the same constant squared. The phase of the reflection coefficient is uniformly distributed between $-\pi$ and $+\pi$.
2. The target reflection coefficient amplitude is the same as in (1) but the phase of the reflection coefficient is a constant between $-\pi$ and $+\pi$ which is specified by the user.
3. The target reflection coefficient amplitude is a constant specified by the user and the phase of the reflection coefficient is uniformly distributed between $-\pi$ and $+\pi$.
4. Both the phase and amplitude of the target reflection coefficients are constants (2) which are specified by the user.

These targets must be arranged in a line according to a selection of internally generated patterns or the target pattern as specified in 'EXTARG'. The input cards for target type selection and target line pattern are described in subsections (B and C) following.

The Physical Along-Track Antenna Pattern

The physical antenna pattern is specified as a function of length along the target line. The user supplies the program with an integer number (CARD 2 columns 11-15) of nominal resolution cells which the physical beam encompasses in the target line. This number is figured by momentarily assuming a physical beam of uniform gain and of the same beamwidth as that which is to be modeled. Such a beam provides a return signal from a point target (assuming C.W. operation) which is a chirp (linearly swept FM) signal of duration T and bandwidth B. It was shown in Chapter III, that the resolution attainable with such a situation is $v_o B^{-1}$ where v_o is the radar system along track velocity. Hence the total number of resolution cells along track in the physical beam of the antenna is $v_o T/(v_o B^{-1})$ or TB. Therefore TB is the number supplied.

The target line contains a specified integral number of targets (CARD 1 columns 21-25 in all cases except when targets are read in from a user supplied tape) and the user has the option of putting an arbitrary integral number of targets in each resolution cell (CARD 2 columns 21-25 for tape supplied targets this number must correspond to that used in generating the targets originally). These will, of course, be equally spaced with respect to each other. The user should be alert to the fact that at the leading and trailing edge of the beam, resolution cells yield π radians of phase shift from one side of the resolution cell to the other and hence one target per resolution cell may not give too accurate results for simulating continuous ground targets.

Two kinds of two-way voltage antenna patterns are provided for the physical beam in the along-track direction. Provision has been made for external input patterns by having the user complete the empty subroutine WGTFCN (Appendix III page 255). The designation of the antenna pattern is done by means of an integer placed on CARD 2 in columns 41-45. The integer 1 calls for a square beam with each target in the physical beam contributing to the return signal in proportion to its voltage reflection coefficient. A "2" designates a two-way pattern which follows the formula:

$$W(x) = rect\left(\frac{x}{a}\right)\left[.3162 + .6838 \cos \frac{\pi x}{a}\right]$$

This is one of a class of functions commonly designated as a cosine on a pedestal. The reader will recall from Chapter II page 24 that the envelope of the spectrum of a linearly swept FM signal closely mimicks the time envelope. Hence the processor response to a point target on the ground (assuming that the processor is doing equi-phase filtering) is the Fourier transform of this envelope. Figure 12 shows the transform pair. The resulting synthetic aperture effective beam on the ground is v_oB^{-1}, where B is the full doppler bandwidth of a return pulse and v_o is the radar velocity. Note that whereas the first side-lobe for a square beam is 13 db down, the first side-lobe for this beam is 20 db down. The point target synthetic aperture responses for antenna patterns "1" and "2" as they are simulated by the computer program are shown in Figures 16 and 17 respectively. For externally supplied antenna patterns, an integer must still be supplied by the user on the data card as indicated in the comments contained in the empty subroutine WGTFCN (Appendix III page 255).

Target Line Pattern Options

There are five sources of target line patterns against which the simulated processor may be tested. The CYCLE pattern is appropriate for checking the frequency response of the processor and is also used for measuring the image signal to noise ratio of the image of a class of targets arising from a single statistical population. To achieve a CYCLE pattern the user writes the word CYCLE on CARD 1 in column 11-15. The approximate total number of point targets in a target line is written as an integer on CARD 1 in columns 21-25. The cycle pattern provides a

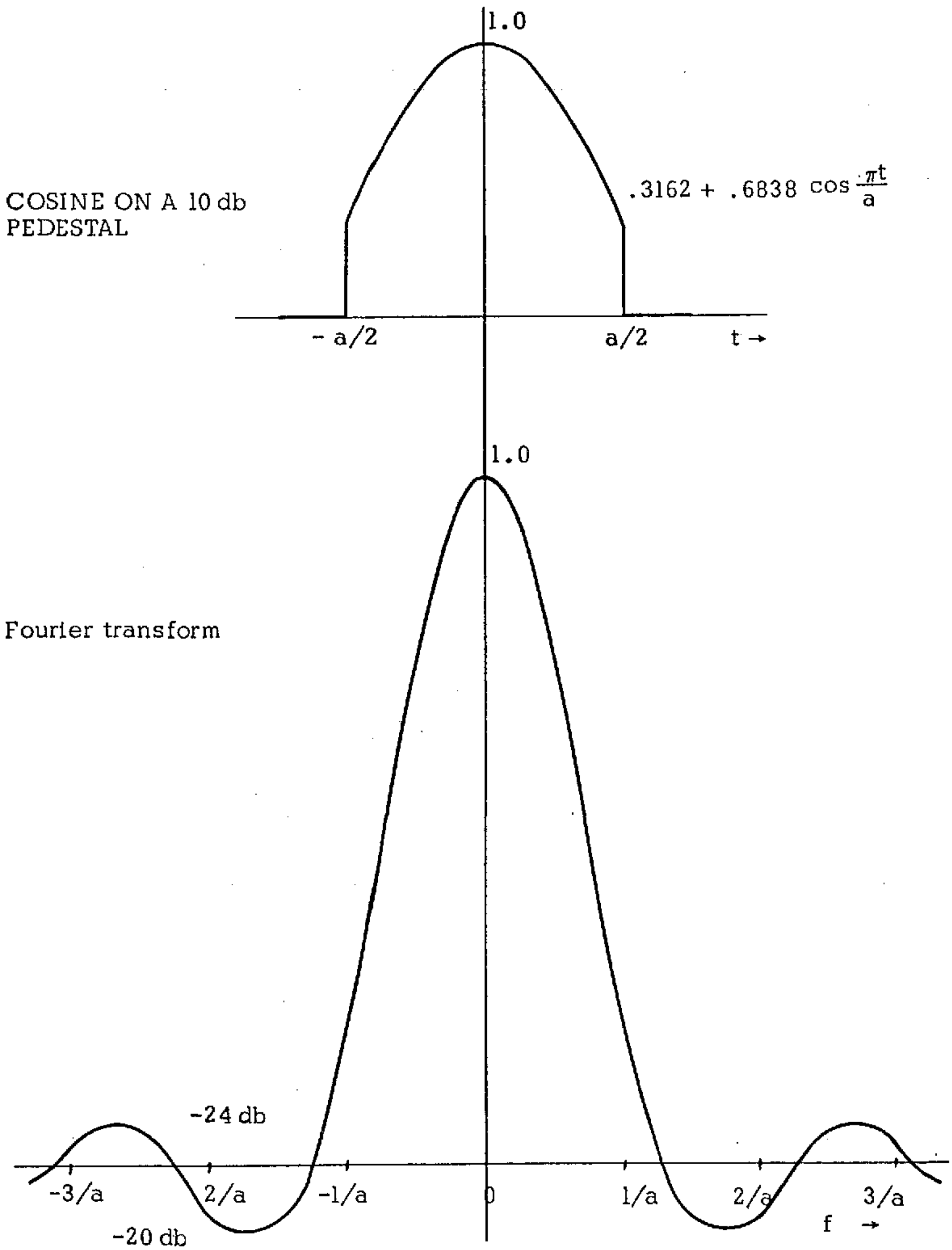


Figure 12. Cosine on a pedestal function used to weight reference functions and/or 2-way antenna pattern in SAR computer simulation program.

target line with alternating sequences of point targets of high and low mean voltage reflection coefficient amplitudes. These amplitudes may be constant numbers (2) or they may come from a square-root chi-square two distribution multiplied by the user-supplied high and low numbers. These numbers are real five digit numbers with two decimal places and they are placed on CARD 1 in columns 40-45 and columns 50-55 respectively. The user will also specify the number of consecutive point targets which are "high" at the beginning of the target line and the number of "low" consecutive point targets at the end of the target line. These integer numbers respectively go on CARD 1 in columns 59-60 and 64-65. The program will then create a target line such that all frequencies of target "highs" and lows are generated beginning with the smaller number in columns 59-60 and ending with the larger number in columns 64-65. At each frequency the number of cycles will be equal. The program automatically increases the total number of targets in the target line to see that this condition is met. For example, had a total target line of 100 targets (CARD 1 columns 21-25) been specified and had columns 59-60 and 64-65 continued the numbers 3 and 5 respectively the program would have generated the following target line: five cycles of 3 "high" point targets in the line and 3 "low" point targets in the line followed by five cycles of 4 "high" point targets in the line followed by 4 "low" point targets in the line followed by 5 cycles of 5 "highs" and 5 "lows". The total target line is 120 targets (rounded upward) rather than the originally specified 100 targets. If the targets are to have constant voltage reflection amplitudes or if they are to come from the voltage Rayleigh distributed population is determined respectively by writing the word CONSTANT on CARD 1 in columns 30-37 or leaving these columns blank. The phase of the voltage reflection coefficients of these targets must also be specified. This is done by writing CPHASE on CARD 1 in columns 69-74 or leaving these columns blank. CPHASE means that all the targets in the line have the same voltage reflection coefficient phase, a real number between $-\pi$ and π with two decimal places in columns 75-80 on CARD 1. If columns 69-74 are left blank the phase of the targets is uniformly distributed between $-\pi$ and $+\pi$ and columns 75-80 on CARD 1 are ignored.

The CYCLE pattern finds a special application when the "high" and "low" targets are set equal and both the phase and amplitude of the reflection coefficients come from the statistical populations which the program generates. When the "high" and "low" targets are set equal (not necessary that either phase and/or amplitudes of target reflection coefficients be statistical) the program automatically increases its output to include a statistical analysis of the reflection coefficient amplitudes of the targets in the line, the return signal voltage amplitude after passage through a possibly noisy narrow band linear preamplifier, and of the image lines (possibly several sub-apertures as discussed in Chapter III) generated. More will be said about this analysis in subsection L of this Chapter. However, because of this feature an integer number must be written respectively on CARD 1 columns 47-48 and on CARD 3 columns 51-55 to determine the sampling interval to be used in the image line and in the sequence of return voltage signals. For example, if the final image line has a resolution of $NB^{-1}v_o$, but the image points are being generated at incremental lengths of v_oB^{-1}, the sampling of the image line for statistical analysis should be no greater than every N^{th} image point to avoid taking correlated samples. One safeguard against this is the fact that the analysis also estimates the degree of sample correlation.

The LINEAR pattern generates a continuously increasing return to measure some of the effects of non-linear system operation. To achieve this pattern the word LINEAR is written on CARD 1 in columns 11-16. The total number of point targets comprising the target line is an integer number on CARD 1 columns 21-25. This pattern does not admit target reflection coeffieint amplitudes from statistical populations. Rather, it requires that the amplitude for the first target in the line be written on CARD 1 in columns 40-45 and that the amplitude of the last target be written on CARD 1 in columns 50-55. Both these numbers are real and have two decimal places. The program divides the difference between the amplitudes of the first and last targets by the total number of targets in the line minus one and increments the reflection coefficient amplitudes of each target in the line sequentially by this number. Columns 69-74 on CARD 1 determine whether the phase of each target reflection coefficient shall be the same or whether the phase is to come from a uniform distribution from $-\pi$ to $+\pi$. If they are to be the same CPHASE is written in those columns otherwise they are left blank. If the phase is to be constant the value of the phase between $-\pi$ and π is written as a two decimal number on columns 75-80 on CARD 1.

The POINT target line pattern is an extremely valuable analytic tool and has been used by the author extensively for curves presented in Chapter V. This pattern gives the processor image of a POINT target as the physical beam sweeps across the point. This respose is essentially a cross section of the processor ambiguity diagram for ν the doppler mismatch equal to zero (see Figure 10b page 40). This pattern is called for by writing POINT on CARD 1 in columns 11-15. If the amplitude of the reflection coefficient is a constant the word CONSTANT is placed in columns 30-37 of CARD 1. The value of that constant is a two decimal real number in columns 40-45 of CARD 1. If columns 30-37 are left blank the amplitude of the reflection coefficient is drawn from a population which is a Rayleigh voltage distribution multiplied by the number in columns 40-45 of CARD 1. The author can see no merit in using this non-constant reflection coefficient; however, it did not seem economic to eliminate the option for a randomly chosen amplitude. The phase of the point target reflection coefficient can also be either a random number between $-\pi$ and π or a definite constant within these limits. For a given constant phase a two decimal real number is written in columns 75-80 on CARD 1 and the word CPHASE must be written in columns 69-74 on the same card. To get a random phase both fields must be left blank.

The user may generate his own target line by completing the dummy subroutine EXTARG according to the instructions contained therein. In this event CARD 1 requires the word EXTARG in columns 11-16 as well as the number of targets in the target line in columns 21-25. In connection with the scheme (described in subsection F of this chapter) for achieving a given signal-to-system-noise ratio at the output of the linear RF preamp, the user may generate a line of targets of a given scattering cross section as a device for finding the average signal power at the receiver. In that event the user will furnish the number of returns incremented between samples of return signal as an integer number on CARD 3 in columns 51-55. This entry flags the program to perform the statistical analysis of return signal. If the user desires a specified signal-to-system-noise at the output of the first preamp the mean signal voltage amplitude on which the ratio is to be based is placed in the form of a real number with two decimal places in columns 60-65 of CARD 3. The signal-to-noise ratio to be achieved goes in columns 41-45 of CARD 3 as a two-decimal real number. Should the user desire a statistical analysis of the image line for externally generated targets, the number of image points incremented between samples from the image line is an integer number in columns 47-48 of CARD 1.

It is possible to store targets and signal return voltages on a user-supplied tape, as described in subsection K of this chapter, and then to use these targets as inputs for runs with other types of processing. If the target line is to be supplied from a tape the word TAPE must be written in columns 11-14 of CARD 1. Since more than one set of targets may be stored on a tape the particular data set number to be used is an integer which goes in columns 21-25 of CARD 1.

Pulse Repetition Frequency

An integer number analogous to the radar P.R.F. is written in columns 41-45 of CARD 4. This is the number of point targets in the line over which the physical beam is moved between signal returns. In theory, for quadrature detection, there must be at least one return measured for each nominal resolution cell distance the radar moves. Thus the maximum number should be equal to the number of targets per nominal resolution cell (CARD 2 columns 21-25). If a tape target input is used the P.R.F. must be the same as that used in creating the tape.

Receiver Gain Function

The program has the ability to demonstrate a reasonably wide variety of kinds of I.F. distortion caused by gain which is a function of the amplitude of the received signal voltage. The general form of the receiver output voltage as a function of input voltage is

$$|V_{OUT}| = |V_{in}| K \qquad \text{for } |V_{in}| \leq D$$

$$|V_{OUT}| = DK + P \log_{10} (|V_{in}| - D + 1)$$

There is no provision made for phase distortion. Much of the time the value of K is left at 1.00 (CARD 3 columns 11-15) and the value of D takes a large value up to 10^8 (CARD 3 columns 16-25) for what in most cases covers the dynamic range of the input. Note that for complete receiver linearity, the amplitude of the return signal voltage with system noise added cannot exceed D. If it does the receiver limits its output to DK. By changing the value of D the effect of an I.F. strip with hard limiting of the signal can be demonstrated. At the same time it is possible to show the effect of soft limiting making the value of P (CARD 3 columns 31-35) something other than zero. In fact if D is zero and P is not zero the program simulates a logarithmic receiver.

Receiver Noise (Appendix II) describes the procedure used to simulate receiver noise in the program. The narrow-band additive Gaussian noise is nominally added to the received signal in the first R.F. preamp. The procedure for adding this noise depends on whether the targets used are internally or externally generated.

With internally generated targets the program goes through one of the four procedures described in Appendix II to determine the mean value of received signal voltage at the antenna based on a mean target voltage reflection coefficient amplitude written on CARD 3 in columns 46-55. The mean value includes the effect of antenna pattern whether it is externally supplied or internally generated. This mean amplitude is squared and then divided by twice the desired signal to system noise ratio (CARD 3 columns 41-45) to yield the variance (σ_N^2) of the population of normally distributed random variables ($N(o,\sigma_N^2)$) which are added to the received signal according to Rice's[19] scheme. Table II shows the results of such noise addition.

The first entry in the table is most illustrative since more than twice as many sample points were used in its determination as were used for the other three target types. This manifests itself in the smaller confidence intervals generally obtained for this target type. The mean power at the preamp output is directly proportional to the square of the mean voltage plus the voltage variance. Before noise was added the measured value of this number was 123. After noise was added to make the signal to system noise equal to 1 this number was measured to be 248 demonstrating that the signal to noise ratio of 1 was indeed achieved.

When an externally generated target line is used, a number equal to the expectation of the square of the amplitude of the signal voltage received at the antenna is written in columns 56-65 of CARD 3 (either EXTARG or TAPE options) along with the desired signal-to-system-noise ratio (columns 40-45 of CARD 3). With this information the program adds the appropriate system noise. However, the problem of determining this number requires another separate run of the program. The user must generate a line of his targets from the same population (i.e., of a single scattering cross-section) and make a statistical analysis of the amplitude of the return signal voltage at the system antenna. Provision has been made so that by inserting the word NONE on CARD 4 in columns 11-14, no processing of these returns will be done and the program will stop after the statistical analysis of the return signal has been completed. The statistical analysis is output in a form quite similar to that used in Table II and the number which is subsequently furnished (CARD 1 columns 40-45) the program to add the proper system noise is the sum of the measured mean voltage squared and the measured voltage variance.

TABLE II

Test Results for Noise Addition Scheme

Target Type: Rayleigh reflection coefficient amplitude with uniformly distributed phase (1000 samples)

	90% Confidence Interval	Confidence Interval Normalized to: a) mean voltage; b) mean voltage squared
No Noise Statistical Results		
Mean voltage measured 9.732	9.01 to 10.545	.926 to 1.08
Voltage variance measured 28.483	25.987 to 30.979	.27 to .33
Sample covariance measured 0.141	-.242 to .524	
Desired Signal to Noise = 1.00		
Mean voltage measured 13.835	12.562 to 15.108	.91 to 1.09
Voltage variance measured 56.174	52.129 to 60.218	.37 to .32
Sample covariance measured 0.089	-0.049 to 0.226	

Target Type: Both phase and amplitude of target reflection coefficients are constant (400 samples)

No Noise Statistical Results		
Mean voltage measured 8.491	8.490 to 8.491	1.0 to 1.0
Voltage variance measured 0	0 to 0	0
Sample covariance measured 1.0	1.0 to 1.0	
Desired Signal to Noise = 1.00		
Mean voltage measured 10.584	6.743 to 14.426	.64 to 1.36
Voltage variance measured 21.743	12.740 to 30.746	.11 to .27
Sample covariance measured -0.059	-1.185 to 1.068	

Target Type: Rayleigh voltage distributed reflection coefficient amplitudes with constant reflection coefficient phases (400 samples)

<u>No Noise Statistical Results</u>		
Mean voltage measured 10.737	9.269 to 12.205	.86 to 1.13
Voltage variance measured 10.450	-2.250 to 23.150	-0.02 to .20
Sample covariance measured -0.064	-0.368 to 0.239	
<u>Desired Signal to Noise = 1.00</u>		
Mean voltage measured 14.973	13.577 to 16.368	.91 to 1.10
Voltage variance measured 55.656	54.983 to 56.328	.28 to .29
Sample covariance measured -0.078	-0.181 to 0.025	

Target Type: Constant voltage reflection coefficient amplitude with uniformly distributed phase (400 samples)

<u>No Noise Statistical Results</u>		
Mean voltage measured 7.020	6.944 to 7.096	.99 to 1.01
Voltage variance measured 13.964	-1.694 to 29.622	-.03 to .61
Sample covariance measured -0.141	-0.159 to -0.123	
<u>Desired Signal to Noise = 1.00</u>		
Mean voltage measured 9.201	3.838 to 14.563	.42 to 1.58
Voltage variance measured 27.468	26.062 to 28.909	.32 to .35
Sample covariance measured -0.021	-0.077 to 0.034	

Doppler Mismatch

In discussing the ambiguity diagram in Chapter II it was determined that the concept in two dimensions, mismatch in time and Doppler frequency was still applicable even though the rate of change of doppler frequency (i.e., radial acceleration) was far from zero. A mismatch in doppler frequency could happen if the physical antenna were boresighted slightly off the normal to the radar bearing vehicle's trajectory (i.e., if the physical antenna were "squinted") or if the local oscillator signal mixed with the return signal after the preamp (see Figure 11) were slightly off. The program makes provision for simulating such an occurrence. One supplies the program with a real number on CARD 2 in columns 31-35 which is the fraction of the total doppler bandwidth of the return signal by which the local osciallator is off. The program then adds the appropriate linearly increasing (in time) phase angle to the return signal voltage.

Return and Reference Signal Quantization

The general digital processing scheme calls for the signal from the I.F. strip to go through an A/D converter immediately before being stored in the digital processor. At this point the signal is bipolar video. The computer program requires an amplitude range for the bipolar video signal and the number of quantization levels into which this range is to be divided in equal increments. The high and low amplitude limits for quantization are written respectively on CARD 7 in columns 31-35 and columns 21-25 respectively. The number of bits available to store the return signal is written on CARD 7 in columns 11-15. Return voltages whose magnitudes are outside this range are truncated to the nearest quantization level. Since the reference or correlation function is also stored in the processor in digital form, it too may be quantized to the number of bits written on CARD 7 in columns 41-45. The program automatically normalizes the reference function to the range between 1 and - 1 and it is this range which is divided equally into these quantization levels.

Focused and Zone Plate Processing Options

The author draws only one distinction between focused and zone plate processing. In focused processing the reference or correlation function mimicks the phase behavior of the return signal from a point target -- it has a quadratic phase factor as a function of time. On the other hand, the reference function for zone plate processing has only two phases, 0 and π. Where the reference function for focused processing will have a factor $\sin kX^2$, the corresponding factor in zone plate processing will be $(\sin kX^2/|\sin kX^2|)$ or $(\cos kX^2/|\cos kX^2|)$. Aside from this difference all the options in one type of processing apply equally well to the other. This difference in factors which go with the reference function in all types of processing which the program implements is called for by putting either the word FULL or ZONE on CARD 4 columns 11-14.

Processing with sub-apertures has been described at length in Chapter III. The number of sub-apertures for this program cannot be less than one (CARD 4 columns 21-25). Each sub-aperture is a given length including so many nominal resolution cells (CARD 4 columns 31-35). The maximum length is the number of nominal resolution cells in the physical antenna's beam. The sub-apertures may overlap by a number of returns. That is to say, the squint angles of the synthetic aperture beams are overlapped. The number which is input to the program (CARD 4 columns 51-55) is either a positive or negative integer giving the number of fully compressed returns which are lapped. The term fully compressed returns denote's the number of pulses returns being used for sub-aperture processing. Thus with no "presumming" this number is the sub-aperture length in nominal resolution cells multiplied by the number of point targets in each resolution cell divided by the number of point targets in each resolution cell divided by the number of targets the physical beam moves per return (PRF). This number is truncated to be an integer. For example, suppose the number of resolution cells in the physical antenna's beam is to be 100 and there are 5 targets per resolution cell and 3 targets are incremented per return and the sub-aperture length is to be 40 nominal resolution cells long. Then with no "presumming" the number of fully compressed returns per sub-aperture is 66. If one wished the synthetic aperture beam angles to overlap by half their beam width the number that would go on CARD 4 in columns 51-55 would be 33.

Presumming alters this situation. The presum option permits one to add each series of N (CARD 6 columns 11-15) returns together, thus contracting the number of effective returns per sub-aperture by the factor N. In the previous example the number of fully compressed returns per sub-aperture for N = 2 would be 33 and the overlap number rather than being 33 would be 16.

It should be realized that the sub-apertures are built around "pieces" of the sampled response from a point target which are used as the correlation or reference function. The reference functions for the full aperture is $\sum_n (t - n\delta) \text{ rect } (\frac{t}{T}) \begin{matrix} \sin \\ \cos \end{matrix} \{kt^2\}$ where T is the time required for the physical beam to sweep by a point target. Sub-apertures are built by performing the required correlations with fractions of these functions, i.e., $\Sigma(t - n\delta) \text{ rect } (\frac{t \pm \alpha}{T_1}) \begin{matrix} \sin \\ \cos \end{matrix} \{kt^2\}$ where $T_1 \leq T$ and $|\alpha| \leq |\frac{T - T_1}{2}|$.
The program offers the user the option of starting his first sub-aperture with any particular compressed reference sample. This is done by putting the numbers of the first reference sample on CARD 4 columns 61-65.

Except for the difference in ZONE or FULL focusing the reference functions have as factors sin kX^2 or cos kX^2 which are sampled to match the fully compressed returns in a sub-aperture for a point target. For example, it was shown in Chapter II that operating under C.W. conditions and with a square physical antenna beam, the return signal from a point target, at zero offset frequency was directly proportional to rect $(\frac{t}{T})$ cos $(kt^2 + \phi)$. If the pulse period were δ then the reference or correlation functions would be $\sum_n (t - n\delta) \text{ rec } (\frac{t}{T}) \begin{matrix} \sin \\ \cos \end{matrix} \{kt^2\}$. For presumming, the same kind of compression (the addition of series of samples) goes on in the reference function as takes place for the signal returns. However, it is possible to weight the reference functions to achieve matched filtering with reduced side-lobes as discussed in Chapter II. Placing a number 1 in columns 71-75 on CARD 4 achieves the standard reference functions $\sum_n (t - n\delta)\ \mathrm{rect}\ (\frac{t}{T})\ {\sin \atop \cos} \{kt^2\}$ as shown in the processing algorithm sketched in Figure 9 page 32. Placing a number 2 in these columns weights the reference functions according to the formula $\sum_n (t - n\delta)\ \mathrm{rect}\ (\frac{t}{T})$ $(.3162 + .6838 \cos \frac{\pi t}{T})\ {\sin \atop \cos} \{kt^2\}$. This substantially reduces the sidelobes of the impulse response (see study of weighted reference functions pages 85 and 107). Provision has been made in the program to input an arbitrary weighting function by completing the empty subroutine WGTFC according to the comments written there (see Appendix III page 255).

The program also affords the user the possibility of weighting the images of the target line generated by each sub-aperture before these images are summed. This is done via the numbers on CARD 5. This provision has been made to accommodate optimum weighting schemes for reducing image variance as detailed by Gerchberg and Haralick[18].

A very useful option in processing is to modify the processing algorithm sketched in Figure 9 page 32 to eliminate the output of 3 of the four correlators replacing their outputs with zeros. This represents non-quadrature detection on a zero offset frequency and essentially degrades the image mean to standard deviation ratio by 3 db as discussed in Chapter III. This is accomplished by putting either a 1 or a 2 on CARD 4 column 20 to indicate respectively quadrature or non-quadrature processing.

Program Output Options

One of the most time-consuming operations in the program is the generation of return signals from the target line. At the same time for purposes of comparing the quality of imagery using different types of processing or a different receiver, etc., the same target returns are used time and again. To prevent wasting computer time, it is possible as detailed, in Appendix III page 167, to place these signals on tape for input in subsequent runs. All that is required in subsequent runs is that the physical antenna be the same number of resolution cells long and that the same number of targets be incremented with the physical beam per pulse return. All the other radar parameters may change. Provision has as well been made to graph on a convenient 8 1/2" by 11" page, point target voltage reflection coefficients, return signal voltage amplitudes and phases, and the final detected image as a function of along-track distance. In addition, the detected image and the target reflection coefficient amplitudes are available as punched cards for subsequent processing and analysis with diverse computer programs. It is also possible to print out virtually every number in all the computer arrays -- clearly labeled as to what they are -- if the user should have some doubt about the results he is achieving. These options are detailed in Appendix III page 177.

Statistical Analysis of Processor Performance

A fundamental basis of these measurements and the conclusions is the central limit theorem which states in effect that for non-pathologic population distributions, sums of uncorrelated samples are normally distributed in the limit with mean and variance equal respectively to the mean and variance of the sampled population multiplied by the number of samples in the sum. Accordingly the program makes two independent sums of the population samples, each sum being one half the number of total samples. The mean estimator is formed by dividing each of these sums by the number of samples in each yielding two random variables each of which is normal with the mean of the underlying distribution and with a variance equal to that of the underlying distribution divided by the number of samples in the sum.

$$VI(1) = \frac{\sum_{i=1}^{n_1} X_i}{n_1} \quad \text{is} \quad \text{normal}\left(\mu, \frac{\sigma^2}{n_1}\right)$$

$$VI(2) = \frac{\sum_{i=n_1+1}^{N} X_i}{n_1} \quad \text{is} \quad \text{normal}\left(\mu, \frac{\sigma^2}{n_1}\right)$$

$$2n_1 = N$$

The mean estimator then is given by:

$$\hat{\mu} = \frac{VI(1) + VI(2)}{2} \quad \text{which is normal}\left(\mu, \frac{\sigma^2}{2n_1}\right)$$

To find the 90 per cent confidence interval for $\hat{\mu}$ a T random variable is formed with one degree of freedom

$$T = \frac{\hat{\mu} - \mu}{\sqrt{\frac{VI(1)^2}{2} + \frac{VI(2)^2}{2} - \hat{\mu}^2}}$$

where μ is unknown

then $P_r(-6.314 < T < 6.314) = .90$ and the 90 per cent confidence interval for μ is given by

$$C.I. = \left(\hat{\mu} - 6.314\sqrt{}, \ \hat{\mu} + 6.314\sqrt{}\right)$$

To estimate the variance a T statistic with one degree of freedom is again employed. As before, the samples are placed in two groups each of which is assumed normal with expectation equal to the true variance of the underlying distribution.

$$V2(1) = \frac{\sum_{i=1}^{n_1} X_i^2}{n_1} - V1(1)^2$$

$$V2(2) = \frac{\sum_{i=n_1+1}^{N} X_i^2}{n_1} - V1(2)^2 \qquad 2n_1 = N$$

The variance estimator is given by

$$\hat{\sigma}^2 = \frac{V2(1) + V2(2)}{2}$$

To find the 90 per cent confidence interval for $\hat{\sigma}^2$ a T random variable is formed with on degree of freedom

$$T = \frac{\hat{\sigma}^2 - \sigma^2}{\sqrt{\frac{V2(1)^2}{2} + \frac{V2(2)^2}{2} - (\hat{\sigma}^2)^2}}$$

and following the previous precedure, the 90 per cent confidence interval is established as

$$C.I. = (\hat{\sigma}^2 - 6.314\sqrt{\ }, \hat{\sigma}^2 + 6.314\sqrt{\ })$$

One of the underlying assumptions in connection with the use of the central limit theorem is that population samples are sufficiently de-correlated (say an absolute covariance no greater than a tenth) that the samples may be regarded as virtually "linearly independent". Therefore, to ensure the goodness of variance and mean estimates output by the program, a covariance estimate of the samples is also made again using the T statistic. That is, the following two statistics are formed:

$$V3(1) = \frac{\sum_{i=1}^{n_1-1} X_i X_{i+1}}{n_1 - 1}$$

$$V3(2) = \frac{\sum_{i=n_1+1}^{N-1} X_i X_{i+1}}{n_1 - 1} \qquad 2n_1 = N$$

The covariance estimator is

$$\hat{\rho} = \frac{1}{2}\left[\frac{V3(1) - V1(1)^2}{V2(1) - V1(1)^2} + \frac{V3(2) - V1(2)^2}{V2(2) - V1(2)^2}\right]$$

To find the 90 per cent confidence interval for $\hat{\rho}$, a T random variable is formed with on degree

$$T = \frac{\hat{\rho} - \rho}{\sqrt{\frac{1}{2}\left[\sum_{j=1}^{2}\left(\frac{V3(j) - V1(j)^2}{V2(j) - V1(j)^2}\right)^2\right] - \hat{\rho}^2}}$$

and the 90 per cent confidence interval for ρ is given by

$$C.I. = (\hat{\rho} - 6.314\sqrt{}\,,\ \hat{\rho} + 6.314\sqrt{}\,)$$

If the confidence intervals computed are too long a larger sample size must be employed. If the samples are too highly correlated a larger sampling interval must be used.

STUDIES IN PROCESSING

Synopsis

This is the primary data section of this paper. It contains curves, tables, and two-dimensional simulated radar pictures to demonstrate various effects which may take place in radar imaging. There is a considerable amount of data on the sub-aperturing techniques detailed in Chapter III.

The chapter is organized so that the first few pages provide information on the principal graphical data format used. Then a series of eighteen studies are described. The graphed data in support of these studies are contained at the end of the chapter.

Introduction

This chapter is devoted to several studies of general types of processing employing the SAR simulation program detailed in Chapter IV. Some of the results are of general interest to those concerned with SAR processing and others are more directly concerned with the investigation of digital processing. However, all the results demonstrate the potential of the software developed over the course of this investigation as a tool for evaluating and simulating a wide variety of SAR processing schemes.

The studies contained in this chapter are illustrated with two-dimensional images, tables of quantitative statistical results and most often by a page of five graphs like that shown in Figure 14 on page 104. To avoid repetitive descriptions of these graphs a detailed discussion of their format will be given here.

The top graph is a plot of point target reflection coefficient phase as a function of position along the image line. The abscissa is the zero phase line and the plot is linear; 0.7 inches above and below the line represents respecitvely phases of π and $-\pi$.

The second graph is a plot of point target reflection coefficient amplitude as a function of position along the image line. The graph is linear in amplitude and is constructed so that the largest amplitude in the target line will always be 1.4 inches above the horizontal axis. The horizontal axis represents zero amplitude. In other words, this graph has an expanding scale. Ordinary SAR systems without provision for doppler frequency averaging will try to reproduce this line as closely as possible. Note that the amplitude of the target reflection coefficient graphed here is proportional to the square root of the target scattering cross section.

The third graph is the phase of the return signal as a function of length along the target line at the output of the R.F. preamplifier relative to a constant clock at the carrier frequency. Of course this makes sense only if the bandwidth of the return signal is very small compared to the carrier frequency. This is the case in the SAR systems under consideration in this report. The graph is laid out just the way the first graph is with a distance of 0.7 inches above or below the abscissa representing respectively π and $-\pi$ radians. The first return signal displayed occurs with the trailing edge of the physical antenna's beam just to the left of the first point target in the target line. Thus this return represents the first return in which only targets in the target line are illuminated by the physical antenna's beam.

The fourth plot is the amplitude of the return signal voltage. Like plot number two, it is on an expanding linear scale so that the largest amplitude is 1.4 inches above the abscissa. Again the first return amplitude in the line comes with the trailing edge of the physical antenna's beam just to the left of the first point target in the target line as in plot number three.

The last plot on the page is the image, generated by the processor, of the target line -- essentially the image of the amplitude of the voltage reflection coefficient. This graph is also on an expanding linear scale so that the largest image voltage amplitude is 1.4 inches above the abscissa.

In all five plots the horizontal axis expands linearly so that the target line is six inches long. Thus any number of targets in the line will always be spread uniformly within the 6 inche abscissa length. Figure 13 will be helpful in visualizing the geometry of the imaging situation represented on the standard plot page.

Study 1 (Figure 14)

Figure 14 shows the performance of a fully focused SAR operating against a cycle target pattern. Neither the phase nor the amplitudes of the reflection coefficients of the targets imaged come from a random distribution. The noise evident in the image line is due to clutter or side-lobes of the synthetic antenna beam. Especially prominent in plot 3 is the fact something radical is happening about two inches from the end of the plot. The disturbance is caused by the fact that the leading edge of the physical antenna's beam reaches the end of the target line here and the program creates reflectionless targets beyond this point.

Study 2 (Figure 15)

Figure 15 shows the performance of the same fully focused radar system operating against a cycle target line pattern. There are two targets per resolution cell, a high followed by a low reflection coefficient amplitude target, at the beginning of the target line. For the remainder of the line there are 2 "highs" followed by 2 "lows" for each cycle. The processor is responding to a cyclic input whose basic spatial frequency

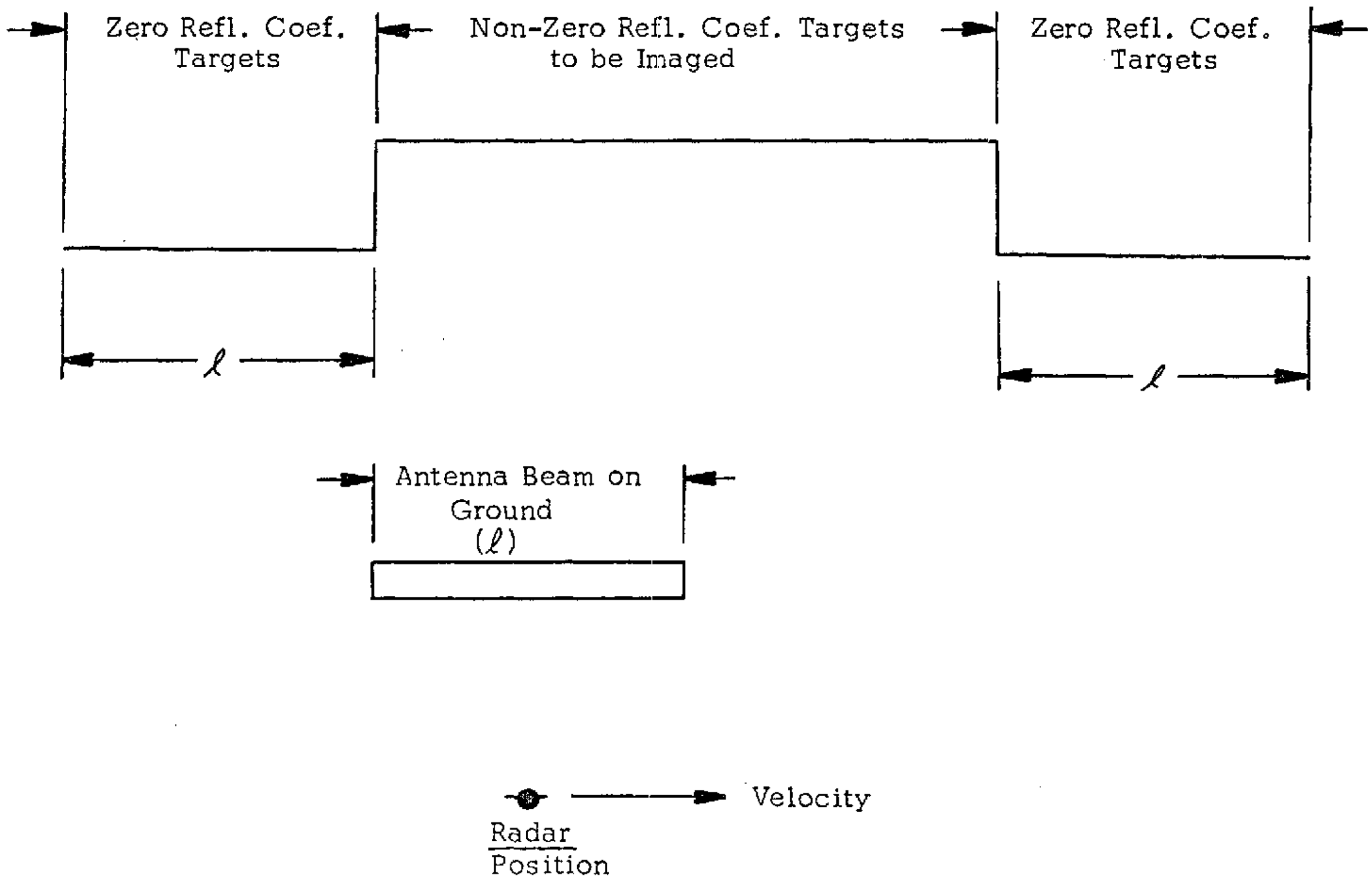


Figure 13. The first two graphs on a standard plot page show the reflection coefficient (1) phase, (2) amplitude. Graphs 3 and 4 show the received signal (3) phase, (4) amplitude, starting with the physical antenna beams in the position shown above. Graphs (5) is the image of the target line generated.

is the inverse of the nominal resolution cell length at the beginning of the target line. The second part of the target line inputs targets at half this frequency. The spatial frequency at the second part of the line represents the nominal spatial frequency of the processor viewed as a filter.

Study 3 (Figure 16)

Figure 16 shows the performance of a fully focused SAR system operating against a point target. The sampling rate (PRF) is 2 1/2 times the Nyquist rate and as a result there are no SAR beam side lobes way off the principal lobe. Note that the principal lobe width is 2 pulses (4 targets) across the top giving the correct half power points for a resolution cell along track about 5 targets long. The first principal side lobes of the SAR beam are about one fifth the height of the principal lobe -- close to the 13.2 db down that simplified theory leads one to expect. The fourth plot in the figure is essentially the picture of the point target that a "brute force" (no processing -- no coherent detection) side looking system would make. The along-track resolution gained by processing is readily apparent. The number of Fresnel zones encountered in sweeping the physical beam by the point target is given approximately by one less than the number of zero crossings in plot 3. That number is 29. The actual number of zones is one half the time-bandwidth product of the C.W. point target return signal on an offset frequency or 30.5.

Study 4 and 5 (Figures 17 and 18)

Using the cosine on a pedestal weighting function on either the reference functions or the antenna pattern reduces the SAR side lobes. The same SAR system as used in Study 3 imaged a point target as shown in Figure 17 when its reference functions were weighted by the program's internally generated cosine on pedestal weighting function. Study 4 still employed a square antenna beam as can be seen by plot 4. The first principle side lobes are approximately one tenth the main lobe height or 20 db down as predicted on the basis of the Fourier transform of the cosine on a pedestal (see Figure 12 page 63). Note that the same image pattern is given when the two-way voltage antenna pattern is the same cosine on a pedestal and the reference functions are flat or unweighted (see Study 5 Figure 18). Furthermore, the processing is immune to the change in phase of the point target reflection coefficient (see plot 1 of Figures 17 and 18). Note too that the amplitude of the return signal shown in plot 4 of Study 5 traces out the cosine on a pedestal antenna pattern as it should. However in both Studies 4 and 5 the price for the reduced side-lobes is a slightly degraded along-track resolution. This is clearly shown by comparing these images to that of Study 3.

Study 6 (Figure 19)

Again in study 6 the same point target and physical beam is employed. However, both the reference functions and the physical antenna pattern are weighted with the cosine on a pedestal function. In this case, because auto-correlation in one domain is the same as squaring the absolute value in the inverse domain one would expect about one per cent side-lobes and that is about what the image shows. However, comparison with studies 3-5 show that the reduction in side-lobe level is paid for with a still slightly larger degradition in along-track resolution.

Study 7 and 8 (Figures 20 and 21)

The nature of zone plate processing as opposed to focused processing is described in Chapters III and IV. Study 7 shows the impulse response for zone plate processing employing the same number of pseudo-Fresnel zones in the reference functions as are employed in full focused processing. Study 7 is directly comparable with Study 3 for full focused processing operating under exactly the same conditions. Note that the close target resolution is virtually the same as for the full focused system and that the first principal side-lobes are also virtually the same. This is what Moore and Rouse[20] predict. The chief difference between the two types of processing comes in the clutter levels in the two types of processing. Zone plate processing will yield an image that will appear somewhat noisier than focused processing under identical conditions because of the increased level of "clutter noise".

Unfocused processing is described in Chapter III. Use of this type of processing against a point target is shown as study 8. The reference functions are only as long as the first Fresnel zone. Aside from this difference, the SAR of Study 8 is directly comparable to the fully focussed and zone plate schemes of Studies 3 and 7 (Figures 16 and 20). The along-track resolution is gross compared to that of these other two studies and the side-lobes are numerous and relatively high making for an image which must appear considerably noisier than either of the other two schemes. The fact that plot 5 appears asymmetric in Study 8 (Figure 21) is due to the fact that the SAR pulses did not come when the point target was directly broadside of the radar bearing vehicle.

Study 9 (Figures 22 ,a,b, and c)

This is a study of the effect of quantizing either or both the reference functions in the processor. The target line pattern employed is a linear ramp in amplitude, but with the phase constant. Figure 22 shows fully focused imaging which is essentially unquantized. The amplitudes of the received signals used in forming the image, range from a low of zero to a high of 800. The line images shown in Figure 22 a indicate that quantization of the reference function does not alter the image much.

In fact Study 7 (Figure 20) on zone plate processing shows that quantizing the reference functions to two levels (1 and −1) does not alter the processor output signal significantly. Figure 22b shows the effect of quantizing the returns alone and Figure 22c shows the effect of quantizing both the returns and the reference functions. The conclusion to be reached is that image breakup is primarily due to quantization noise added to the return signal.

Study 10 (Figure 23)

Figure 23 shows the effect of presumming first 2 and then 3 pulse returns. As anticipated in Chapter III, the resolution appears degraded in direct proportion to the number of pulses presummed. Figure 23 shows only the image line generated by the processor, inasmuch as the target lines and the return signals are the same as those shown in graphs 1-4 in Study 8 (Figure 21).

Study 11 (Figures 24 and 24a)

Figures 24 and 24a show the effect of Doppler mismatch on the processor response as the SAR operates against a point target. The progressive doppler mismatch point images correspond to cross sections on the ambiguity diagram of Figure 10b on page 40 for increasingly larger values of ν. Physically this effect would be produced if the sum of the STALO and COHO frequencies differed from the radar carrier frequency by ν (see Figure 11).

Study 12 (Figures 25)

This study shows the effect of a receiver with a non-linear gain characteristic. The receiver has been "fixed" to hard limit at a value approximately 1/2 the maximum return signal voltage. The spike shown on the end of the image line is geniune. It occurs because the computer simulates a line of zero reflection coefficient targets beyond the end of the target line. Hence the last return signals are small and unlimited allowing an undistorted output signal of the last few image line targets.

Study 13 (Figures 26 and 26a)

This is a study of 5 sub-apertures operating against a point target. Figure 26 is the sum of the five sub-apertures. It is readily seen that the resolution is five nominal resolution cells. Each horizontal movement of the pen in this case corresponds to a nominal resolution length had the processor been operating fully focused. Each of the 5 sub-apertures whose separate images of the point target are shown in Figure 26a employs non-overlapping fifths of the complete reference functions (which would be employed were the processing fully focused). The time-bandwidth product of a C.W. signal from the point target as the physical antenna beam sweeps by it is 375 hence each sub-aperture has a time-bandwidth product of 15. This study is made for quadrature detection.

Study 14 (Figures 27 and 27a)

The same SAR system of Study 11 is used against a point target. The difference in the processing is that only one correlation of the four required is used (see quadrature processing algorithm Figure 9 page 32). The figures are for non-quadrature subaperture processing. Figure 27a shows the sum of the subapertures. This will change as a function of the phase of the point target reflection coefficient. However, the close target resolution will always be about the same, as if the processing were full focused and the response will always have a more or less uniformly distributed clutter level (sometimes referred to as a "thumb tack" response). As a function of phase, the main response lobe will change its amplitude and may even split into two or more lobes. The prominent clutter levels in each of the subaperture responses (Figure 27a) originates in the square law detection as detailed in Chapter III.

Study 15 (Figures 28 and 28a)

This is a basic study in image quality enhancement by using subaperture processing. Figure 28 shows the image line generated by using only one of four possible subapertures to create an image of Rayleigh targets of a single scattering cross-section. Figure 28a shows the same image line but employs 4 non-overlapping subapertures to generate the image line. Since the TB products of the subapertures generated in the two figures are the same (TB = 6.25), the along-track resolution of each of the two image lines is also the same. If the images were ideal, the image lines would be constant, indicating that the target line was made up of targets of a single scattering cross section. One can see that this ideal is much more closely approached with 4 subapertures (Figure 28a) than with 1 (Figure 28).

In Table III below a statistical analysis of processing performance against this target line is given. The image figure of merit, the M/STD ratio, is shown to be largest for that processor employing the four subapertures. Note that decreasing the along-track resolution does nothing to enhance the M/STD ratio of the image line.

Study 16 (Table IV)

This study shows the effects of quadrature as opposed to non-quadrature processing for Fully Focused processing, operating against

TABLE III

Statistical Analysis of Target and Image Lines Showing the Effect of Resolution and Subaperture Processing

	90% Confidence Interval	Confidence Interval Normalized to: a) mean; b) mean squared
Target Line Analysis		
Mean reflection coefficients amplitude measured 1.247	1.085 to 1.410	.87 to 1.13
Reflection coefficients amplitude variance measured 0.444	0.434 to 0.454	.28 to .29
Sample covariance measured 0.033	-0.291 to 0.357	
M/STD measured $\sqrt{3.502}$		
Image Line Analysis - Fully Focused Processing, TB = 100, Along-Track Resolution = L		
Mean image voltage measured 129.0	103.8 to 156.1	.80 to 1.21
Image voltage variance measured 4903	4110 to 5696	.25 to .35
Sample covariance measured 0.035	-0.43 to -0.5	
M/STD measured $\sqrt{3.446}$		
*Image Line Analysis - One Subaperture Employed with TB = 6.25, Along-Track Resolution = 4L		
Mean image voltage measured 66.48		
Image voltage variance measured 1101.3		
M/STD measured $\sqrt{4.0}$		

*Image Line Analysis - Four Subapertures Employed each with TB = 6.25
Along-Track Resolution = 4L

Mean image voltage measured 255.2

Image voltage variance measured 4587.3

M/STD measured $\sqrt{14.2}$

*These readings were made before the final version of the statistical meter was incorporated in the computer program and hence they are imcomplete.

Rayleigh distributed targets. The synthetic apertures which are used, are those of Studies 13 and 14, respectively. Figures 26 and 27 show the synthetic antenna patterns. The obvious conclusion to be reached on the strength of Table IV is that non-quadrature detection decreases image quality as measured by the M/STD ratio by 3 db. However, whether either quadrature or non-quadrature detection is used, the M/STD of the image increases in direct proportion to the square root of the number of non-overlapping subapertures employed in processing. It will be noted in the table that some fairly large confidence intervals are encountered. This comes about because the results are based on only 100 samples. However, the thrust of these results seems to be incontravertable.

Study 17 (Figure 29 and 29a)

Based on the results of the quadrature and non-quadrature subaperture processing studies in hand, it was decided to manufacture a two-dimensional radar image. The method chosen was to "fly" 15 separate target lines of 90 targets and to "stack" the image lines generated, raster

TABLE IV

Statistical Analysis for Quadrature and Non-quadrature Imaging of Rayleigh Targets using Non-overlapping Subaperture Processing

	90% Confidence Interval	Confidence Interval Normalized to: a) mean, b) mean squared
<u>Target Line</u>		
Mean reflection coefficient amplitude measured 1.222	0.212 to 1.231	.99 to 1.01
Reflection coefficient amplitude variance measured 0.433	0.421 to 0.465	.31 to .33
Sample covariance measured 0.090	0.000 to 0.181	
M/STD measured $\sqrt{3.371}$		
<u>First Subaperture Image Line with Quadrature Detection</u>		
Mean image voltage measured 210	174 to 247	.83 to 1.18
Image voltage variance measured 12451	7511 to 17389	.17 to .40
Sample covariance measured 0.193	-0.41 to 0.79	
M/STD measured $\sqrt{3.553}$		
<u>First Subaperture Image Line with Non-Quadrature Detection</u>		
Mean image voltage measured 70	31 to 109	.44 to 1.56
Image voltage variance measured 3728	-693 to 8150	-.14 to 1.66
Sample covariance measured 0.096	-0.01 to 0.20	
M/STD measured $\sqrt{1.327}$		

Second Subaperture Image Line with Quadrature Detection

Mean image voltage measured 204	143 to 265	.7 to 1.3
Image voltage variance measured 12708	9176 to 16239	.22 to .39
Sample covariance measured .001	-0.86 to 0.87	
M/STD measured $\sqrt{3.293}$		

Second Subaperture Image Line with Non-Quadrature Detection

Mean image voltage measured 89	22 to 156	.25 to 1.75
Image voltage variance measured 4517	-1071 to 10,107	-1.3 to 1.27
Sample covariance measured 0.00	-0.36 to 0.37	
M/STD measured $\sqrt{1.766}$		

Third Subaperture Image Line with Quadrature Detection

Mean image voltage measured 221	183 to 259	.83 to 1.17
Image voltage variance measured 14656	3154 to 26,157	.07 to .53
Sample covariance measured -0.22	-0.71 to 0.26	
M/STD measured $\sqrt{3.340}$		

Third Subaperture Image Line with Non-Quadrature Detection

Mean image voltage measured 96	59 to 133	.62 to 1.39
Image voltage variance measured 6211	-6468 to 18891	-.70 to 2.10
Sample covariance measured 0.00	-0.37 to 0.38	
M/STD measured $\sqrt{1.499}$		

Fourth Subaperture Image Line with Quadrature Detection

Mean image voltage measured 198	187 to 208	.95 to 1.05
Image voltage variance measured 14408	-1762 to 30,579	-.04 to .77
Sample covariance measured 0.04	-0.08 to 0.17	

M/STD measured $\sqrt{2.728}$

Fourth Subaperture Image Line with Non-Quadrature Detection

Mean image voltage measured 86	-6 to 179	-0.07 to 2.08
Image voltage variance measured 3496	510 to 6475	0.07 to 0.87
Sample covariance measured 0.00	-0.16 to 0.17	

M/STD measured $\sqrt{2.123}$

Fifth Subaperture Image Line with Quadrature Detection

Mean image voltage measured 196	88 to 303	0.45 to 1.55
Image voltage variance measured 12,670	8204 to 17135	0.21 to 0.45
Sample covariance measured 0.00	-1.49 to 1.49	

M/STD measured $\sqrt{3.044}$

Fifth Subaperture Image Line with Non-Quadrature Detection

Mean image voltage measured 64	19 to 109	0.3 to 1.70
Image voltage variance measured 2327	2162 to 2492	0.53 to 0.61
Sample covariance measured -0.01	-0.83 to 0.80	

M/STD measured $\sqrt{1.810}$

<u>Summation of 5 Subaperture Image Lines with Quadrature Detection</u>

Mean image voltage measured 1030	993 to 1067	0.96 to 1.04
Image voltage variance measured 73898	60629 to 87168	0.06 to 0.08
Sample covariance measured 0.13	-0.14 to 0.40	

M/STD measured $\sqrt{14.378}$

<u>Summation of 5 Subaperture Image Lines with Non-Quadrature Detection</u>

Mean image voltage measured 407	338 to 478	0.83 to 1.17
Image voltage variance measured 19116	8731 to 29501	0.05 to 0.17
Sample covarinace measured 0.02	-0.34 to 0.39	

M/STD measured $\sqrt{8.673}$

fashion, to form a two-dimensional image. Each line was composed of two "classes" of targets which butted against each other at the appropriate point in the target line so that when all 15 lines were stacked there would be a diagonal boundary across the picture. Figure 29 (1) shows a map of the target scattering cross-sections (proportional to the mean voltage reflection coefficient amplitude squared) in two dimensions. To make this display possible R. M. Haralick's[21] program "PITCHR" was used. Punched cards (which were generated in the SAR simulation program) containing the target and image line voltage provided the input to PITCHR. PITCHR allowed the original 15 by 90 point targets to be expanded along both the x and y axes. In so doing, the program performed a two-dimensional interpolation to find values for the new points added to the map. Hence, Figure 29 (1) shows a gradual change in cross section across the diagonal boundary where in fact the boundary changed instantly from one point to the next. Figure 29 (2) shows the imaging that is achieved by Full Focused processing such that each of the original ninety point targets in a line occupies a nominal along-track resolution length. For purposes of illustration the targets used to generate Figure 29 were <u>not</u> statistically distributed. The somewhat gradual boundary image in Figure 29 (2) again is due to interpolation by PITCHR and not to degraded resolution by the imager. Figure 29 (3) takes the same image in plate (2) but narrows the quantization range of PITCHR. The actual range of image voltages in the image line extended from 100 to 1100. Figure 29 (2) had this range quantized into 13 equally spaced grey levels. Figure 29 (3) has the range 800 to 850 quantized into the 13 levels. Figure 29 (3) represents a very limited dynamic range display device as such it is an extreme example of the power of the display device in altering the final image. Figure 29 (2) shows lobing effects of the synthetic aperture beam expecially close to the beginning and end of the image in the along-track direction. For purposes of imaging, beyond either edge in the along-track direction there was zero return voltage (no targets).

Figure 29a shows the results of supaperture processing using Rayleigh distributed targets in the lines. Figure 29a(1) shows a map of the amplitudes of the voltage reflection coefficients and is essentially the image that would be generated by Full Focused processing where each point target was separated by an along-track resolution length. For subaperture processing the along-track resolution length has been degraded by 5 (5 non-overlapping subapertures are used to image). Figure 29a(2) shows the image with Non-Quadrature subaperture processing and Figure 29a (3) shows the image with Quadrature subaperture processing. The images in Figure 29a, demonstrate the utility of the M/STD ratio as an image figure of merit. The three plates in the figure have M/STD ratios of 1.9, 3.0 and 4.25 respectively. Because the along-track resolution in plate (1) is one fifth that of plates (2) and (3) the grey tone resolution of plate (1) and (5) are equal if one "stands back" from the plate (1). Obviously plate (3) is to be preferred to plate (2) for its higher tonal resolution (plates (2) and (3) have equal resolution lengths).

Study 18 (Figures 30, and 30(a)

This study shows the effect that an arbitrary phase factor in the voltage reflection coefficient of a point target can have on non-quadrature processing. The first image of the point target shown as the fifth plot in Figure 30 is for the reference function matched to the return signal. The amplitude of the main response lobe on an arbitrary scale is 30. The extensive clutter responce is predicted in Chapter III. This image is the auto-correlation of a V-FM signal.

The top plot in Figure 30(a) is for the same point target but with a reflection coefficient phase of $\pi/4$ relative to the point target imaged in Figure 30. The height of the main response lobe is 21.2 compared to 30. The clutter level is approximately the same in both plots taking into account the expanding vertical scale.

The bottom plot in Figure 30 (a) shows the image of the same point target with $\pi/2$ phase shift. This plot is the cross correlation of two V-FM signals with one shifted 90 degrees with respect to the other. The highest value in the plot is 4.2.

The point of this study is that non-quadrature processing is drastically effected by the voltage reflection coefficient phase of targets being imaged. Therefore, with targets which have a uniformly distributed phase, one would anticipate a lower M/STD ratio, than with quadrature processing. In fact it appears that the penalty for non-quadrature processing is about a 3 db loss in M/STD.

STUDY FIGURE CAPTIONS

Figure 14 Study 1.	Full Focused SAR operating against a CYCLE pattern of targets. Synthetic Aperture TB product is 100; 1 target per nominal resolution cell length; pulse period every target; flat antenna gain and reference functions; quadrature processing.
Figure 15 Study 2.	Full Focused SAR operating against a CYCLE pattern of targets. Synthetic Aperture TB product is 100; 2 targets per nominal resolution cell length; pulse period every target; flat antenna gain and reference functions; quadrature processing.
Figure 16 Study 3.	Full Focused SAR operating against a point target. Synthetic Aperture TB product is 61; 5 targets per nominal resolution cell length; pulse period every second target; flat antenna gain and reference functions; quadrature processing.
Figure 17 Study 4.	Full Focused SAR operating against a point target. Synthetic Aperture TB product is 61; 5 targets per nominal resolution cell length; pulse period every second target; flat antenna gain; reference functions weighted with a "cosine on a pedestal;" quadrature processing.
Figure 18 Study 5.	Full Focused SAR operating against a point target; Synthetic Aperture TB product is 61; 5 targets per nominal resolution cell length; pulse period every second target; both the two-way voltage antenna pattern and the reference functions are weighted by a "cosine on a pedestal;" flat reference functions; quadrature processing.
Figure 19 Study 6.	Full Focused SAR opertating against a point target; Synthetic Aperture TB product is 61; 5 targets per nominal resolution cell length; pulse period every second target; both the two-way voltage antenna pattern and the reference functions are weighted by a "cosine on a pedestal;" quadrature processing.
Figure 20 Study 7.	Full Zone plate processor SAR operating against a point target; Synthetic Aperture TB product is 61; 5 targets per nominal resolution cell length; pulse every second target; flat antenna gain and reference functions; quadrature processing.

Figure 21 Study 8. Unfocused SAR operating against a point target. The Synthetic Aperture TB product is 2; 5 targets per nominal resolution cell length; pulse period every second target; flat antenna gain and reference functions; quadrature processing.

Figure 22 Study 9. Quantization study, Full Focused SAR operating against a linear ramp target line. Synthetic Aperture TB product is 60; 6 targets per nominal resolution cell length; pulse period every third target; flat antenna gain and reference functions; quadrature processing.

Figure 22(a) Study 9. Image lines only, parameters same as in Figure 22 but reference functions ranging from 1 to -1 are quantized to 5 bits in the first plot, 4 bits in the second plot, and 3 bits in the third plot.

Figure 22(b) Study 9. Image lines only, parameters same as in Figure 22 but voltages of signals in quadrature channels ranging from -800 to 800 are quantized in the first plot to 4 bits; and in the second plot to 3 bits.

Figure 22(c) Study 9. Image lines only, parameters same as in Figure 22 but both are return signal voltages and reference functions are quantized in the first plot to 5 bits, in the second plot to 4 bits and in the third plot to 3 bits.

Figure 23 Study 10. Presumming study showing image lines only. Parameters are the same as in Figure 21. First plot shows presumming two return pulses and the second plot shows presumming three.

Figure 24 Study 11. The effect of doppler mismatch on the point target response of a Full Focused SAR. Synthetic Aperture TB product is 100; 1 target per nominal resolution cell length; pulse period every target; flat antenna gain and reference functions; quadrature processing; doppler mismatch is 3.3% return signal bandwidth.

Figure 24 (a) Study 11. Image lines only-Parameters same as in Figure 24. The first plot shows doppler mismatch to be 6.7% of the return signal bandwidth, the second plot is 50% and the third plot is 75%.

Figure 25 Study 12. The effect of hard limiting in the mid-range of the received signal from a linear ramp target line. Full Focused SAR; Synthetic Aperture TB product is 61; 1 target per nominal resolution cell length; pulse period every target; flat antenna gain and reference functions.

Figure 26 Study 13. Subaperture processing against a point target. The Full Focused Synthetic Aperture TB product is 375. Each synthetic subaperture has a TB product of 15; 1 target per nominal resolution cell length (on the basis of the Full Focused aperture); pulse period every target; flat antenna gain and reference functions; quadrature processing. The image line is the sum of the five subaperture image lines shown in Figure 26a.

Figure 26 (a) Study 13. Parameters same as in Figure 26. This figure shows image lines only of the five non-overlapping synthetic subaperture summed to create the image line of Figure 26. Each subaperture has a TB product of 15 and uses one fifth the bandwidth of the return signal spectrum.

Figure 27 Study 14. Subaperture processing against a point target. The Full Focused Synthetic Aperture TB product is 375. Each synthetic subaperture has a TB product of 15; 1 target per nominal resolution cell length (on the basis of Full Focused processing); pulse period every target; flat antenna gain and reference functions; non-quadrature processing. The image line is the sum of the five subaperture image lines shown in Figure 27a

Figure 27 (a) Study 14. Parameters same as in Figure 27. This figure shows image lines only of the five non-overlapping synthetic subapertures summed to create the image line of Figure 27. Each supabperture has a TB product of 15 and uses one fifth the bandwidth of the return signal spectrum. The processing is non-quadrature.

Figure 28 Study 15. Demonstration of the effect that subaperture processing has on the variance of the image of a Rayleigh distributed line of targets. Subaperture processing using only one synthetic subaperture using one fourth the bandwidth of the return signal. The TB product of the subaperture is 6.25 (for Full Focusing it would be 100); 1 target per nominal resolution cell length (based on Full Focusing); pulse period every target; flat antenna gain and reference functions; quadrature processing.

Figure 28 (a) Study 15. Parameters same as in Figure 28. The image line is the sum of four non-overlapping subapertures. The reduced variance in the image line compared to that of Figure 28 is evident.

Figure 29 Study 17. Fully Focused SAR imaging over a boundary separating two fields whose non-statistically distributed scattering cross sections differ by 9.54 db. (1) "True" map of fields, (2) SAR image with the dynamic range of the display matched to the image dynamic range, (3) SAR image with picture dynamic range much smaller than image range.

Figure 29 (a) Study 17. (1) "True" map of Rayleigh distribution scattering cross sections for two fields whose differential scattering cross sections differ by 9.54 db. - M/STD $=\sqrt{3.6}$; (2) non-quadrature subaperture processing - 5 subapertures - M/STD $=\sqrt{9}$; Quadrature subaperture processing - 5 subapertures - M/STD $=\sqrt{18}$.

Figure 30 Study 18. The effect of Full Focused non-quadrature processing on a point target with various reflection coefficient phases. The TB product is 60; 1 target per nominal resolution cell length; pulse period every target; flat antenna gain and reference functions; non-quadrature processing. The processor reference function and the return signal from the point are matched. The amplitude of the image main lobe is 30.

Figure 30(a) Study 18. This figure shows image lines only. The parameters are the same as for Figure 30. The first image is for the phase of the reflection coefficient shifted 45 degrees relative to that used in Figure 30, and the second is for a phase shift of 90 degrees. The maximum relative amplitudes of the image lines are 21.2 and 4.2 respectively (recall that the images are on expanding scale).

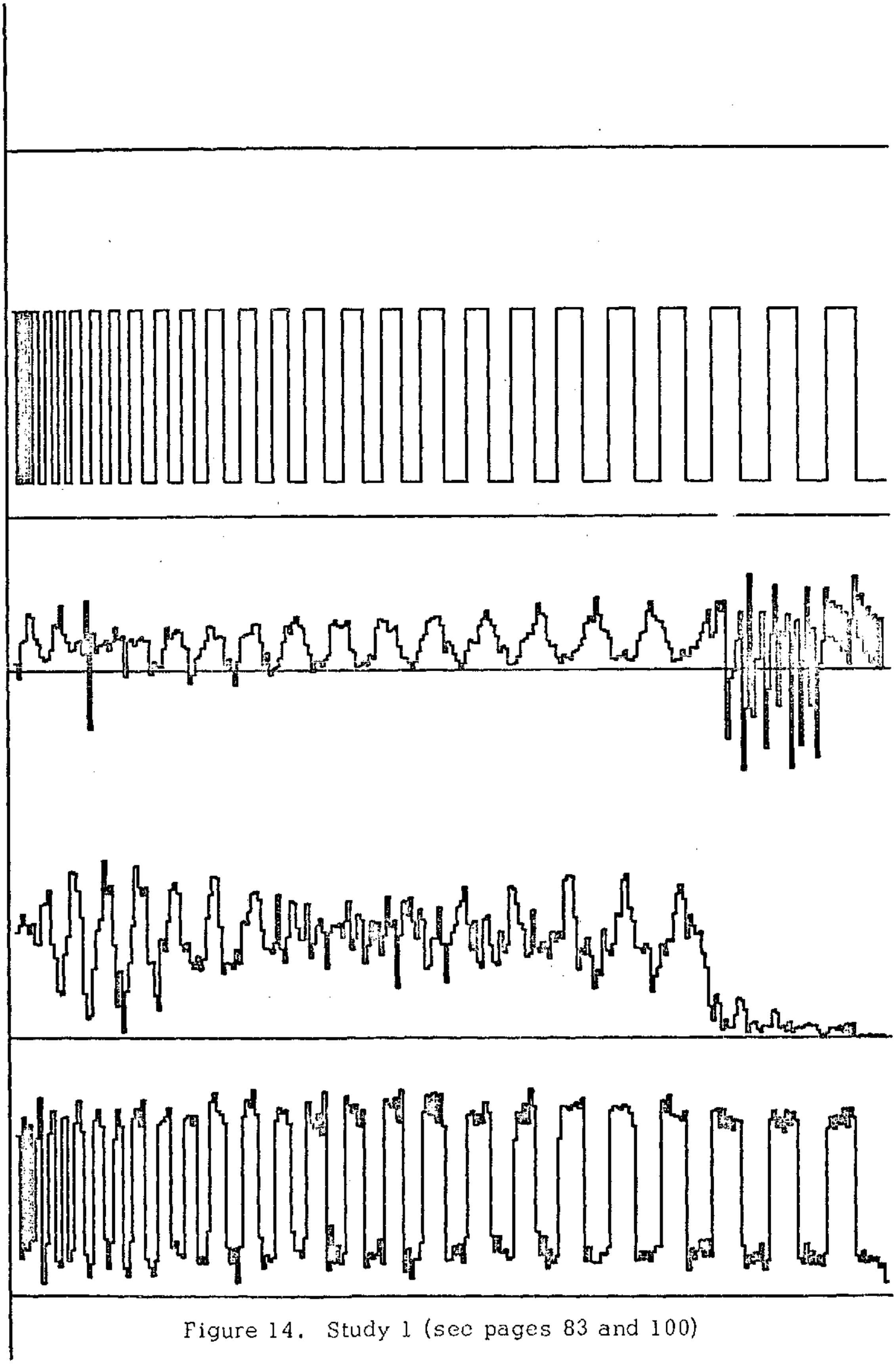


Figure 14. Study 1 (see pages 83 and 100)

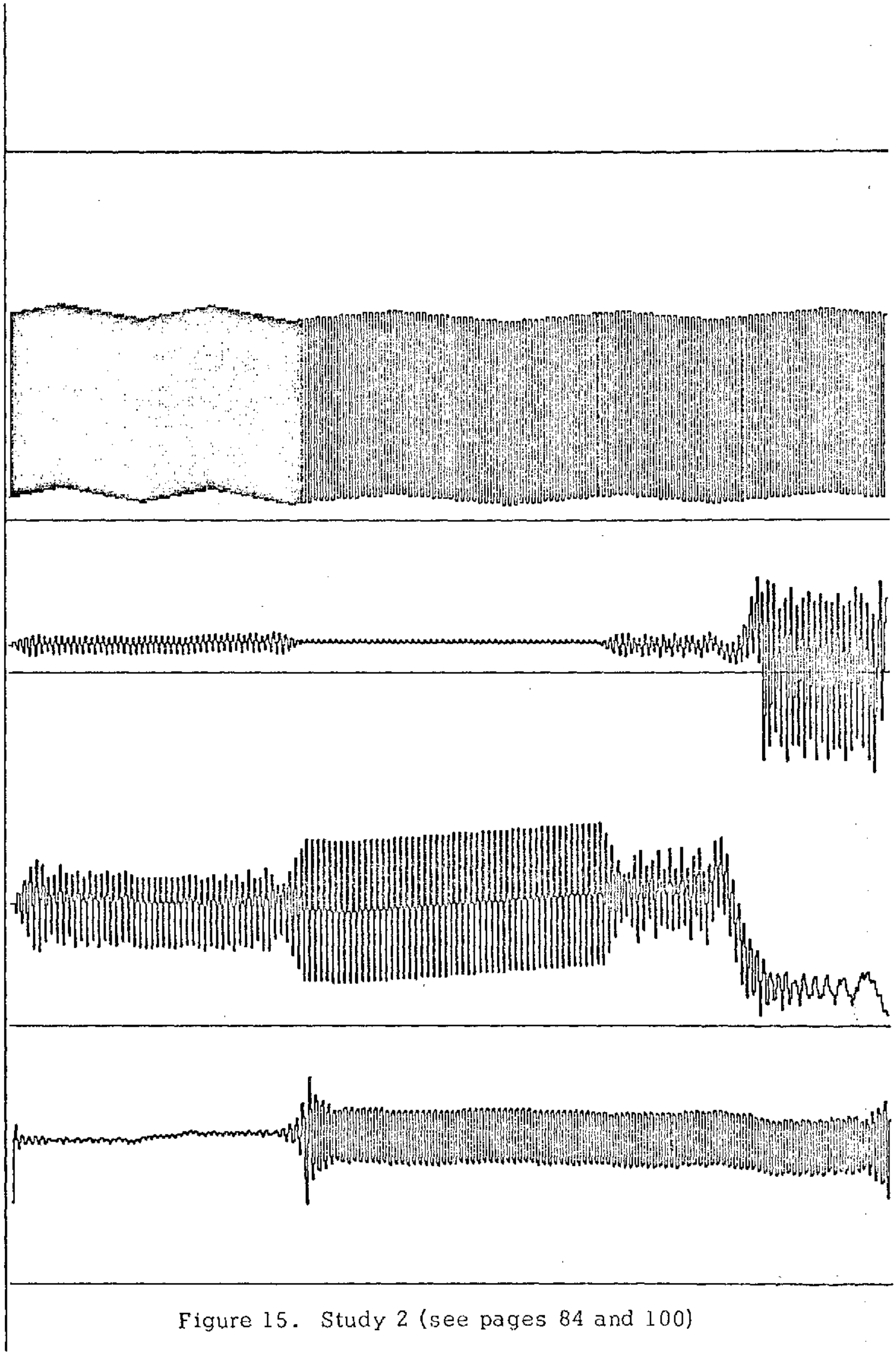


Figure 15. Study 2 (see pages 84 and 100)

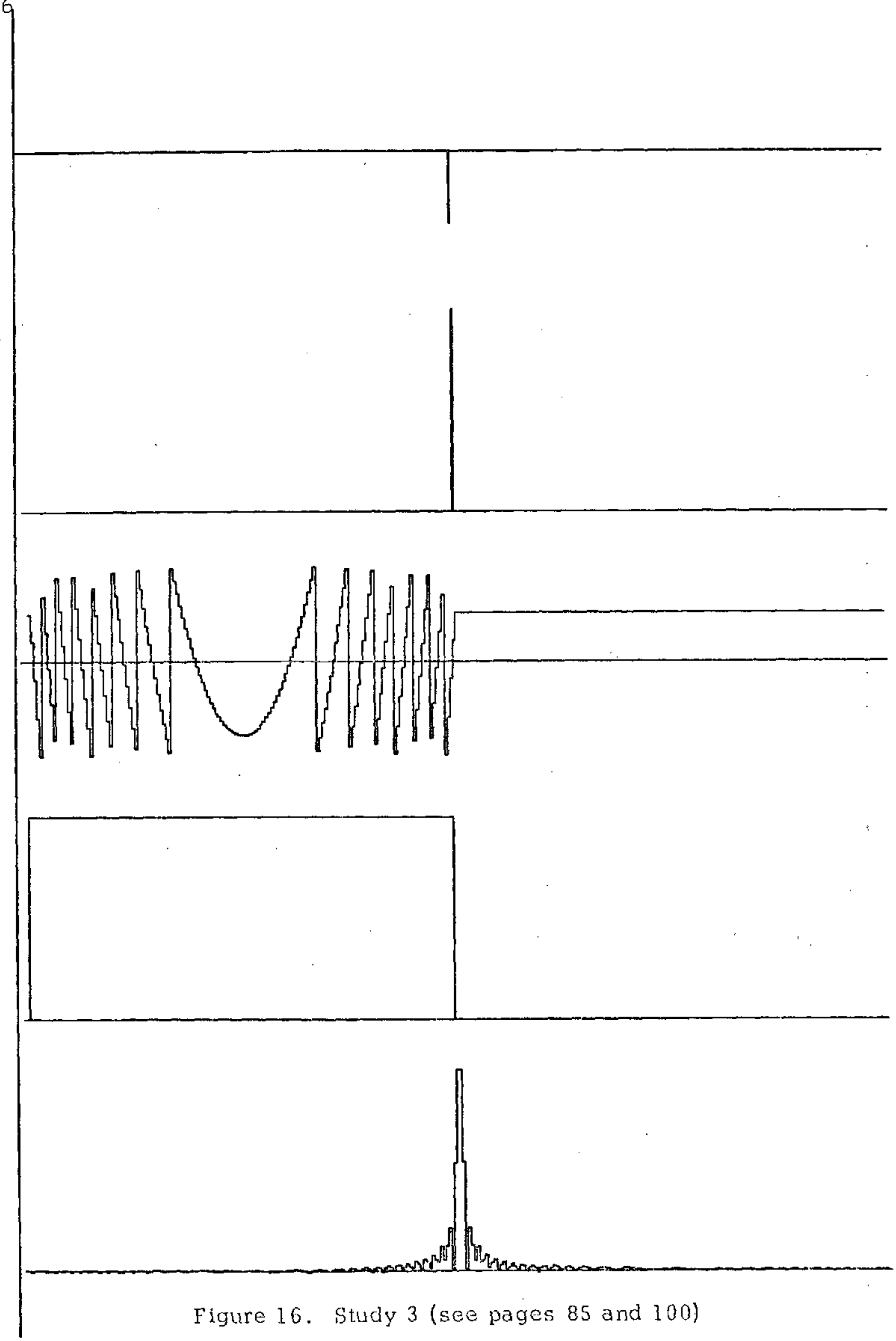


Figure 16. Study 3 (see pages 85 and 100)

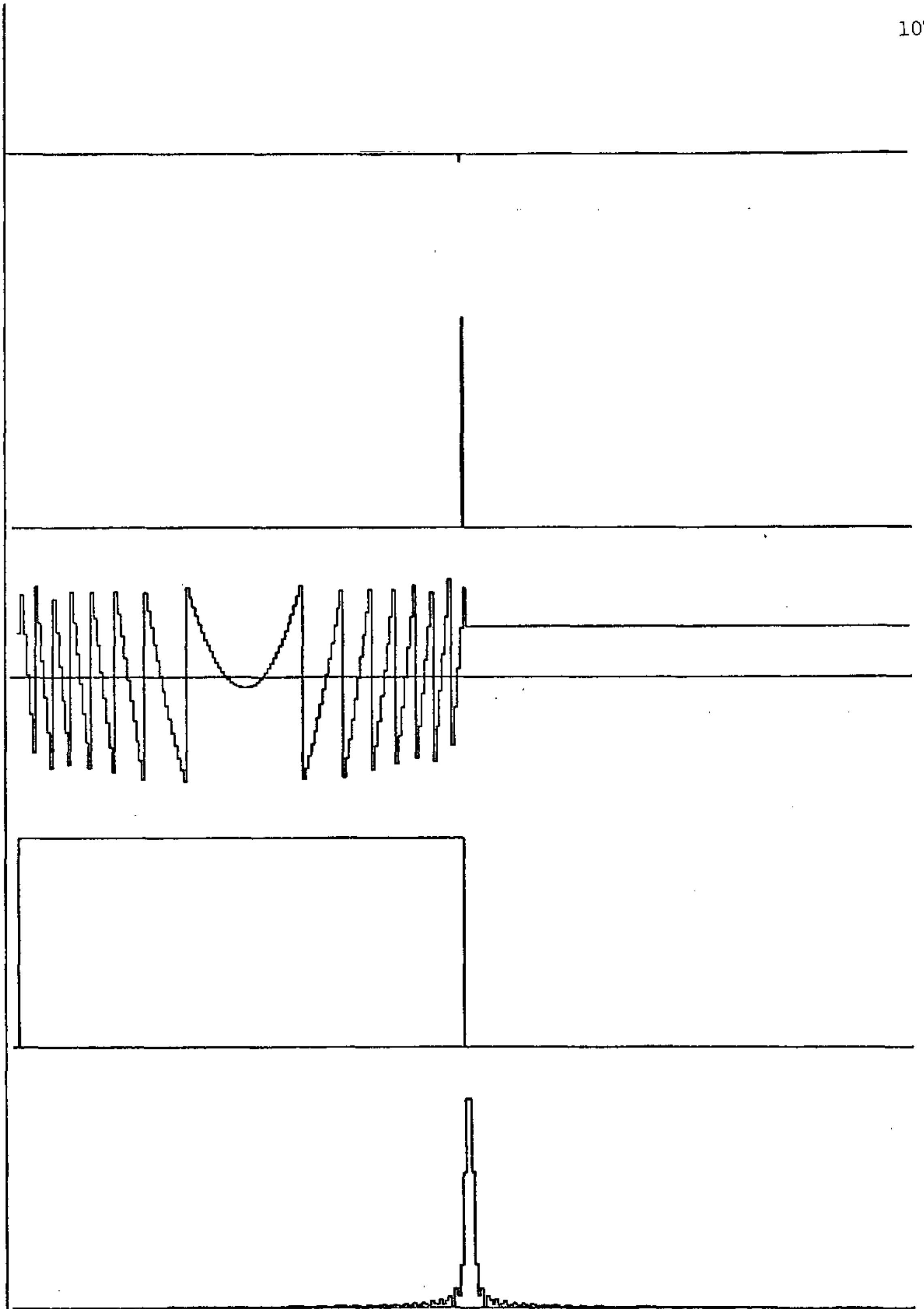


Figure 17. Study 4 (see pages 85 and 100)

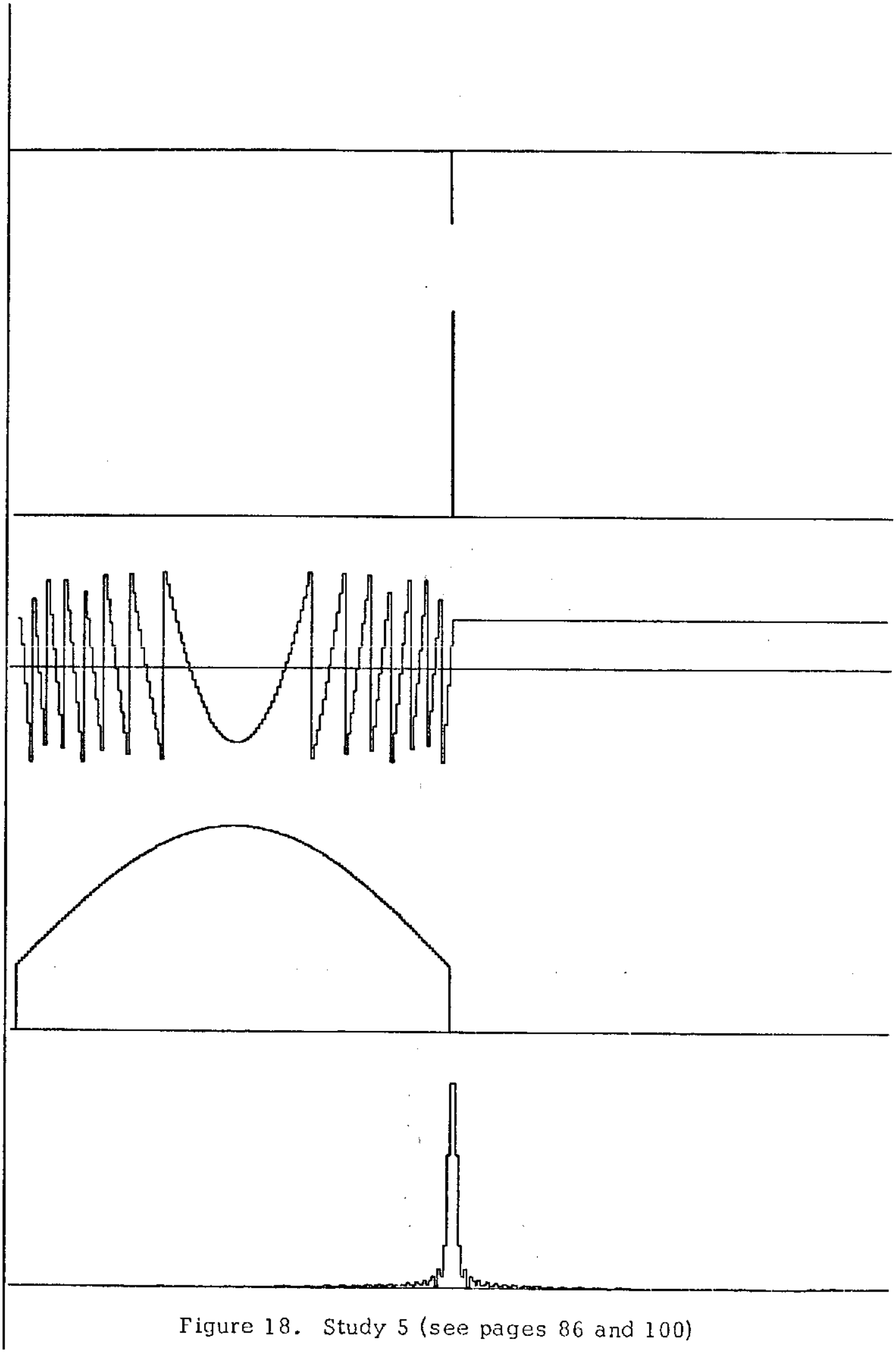


Figure 18. Study 5 (see pages 86 and 100)

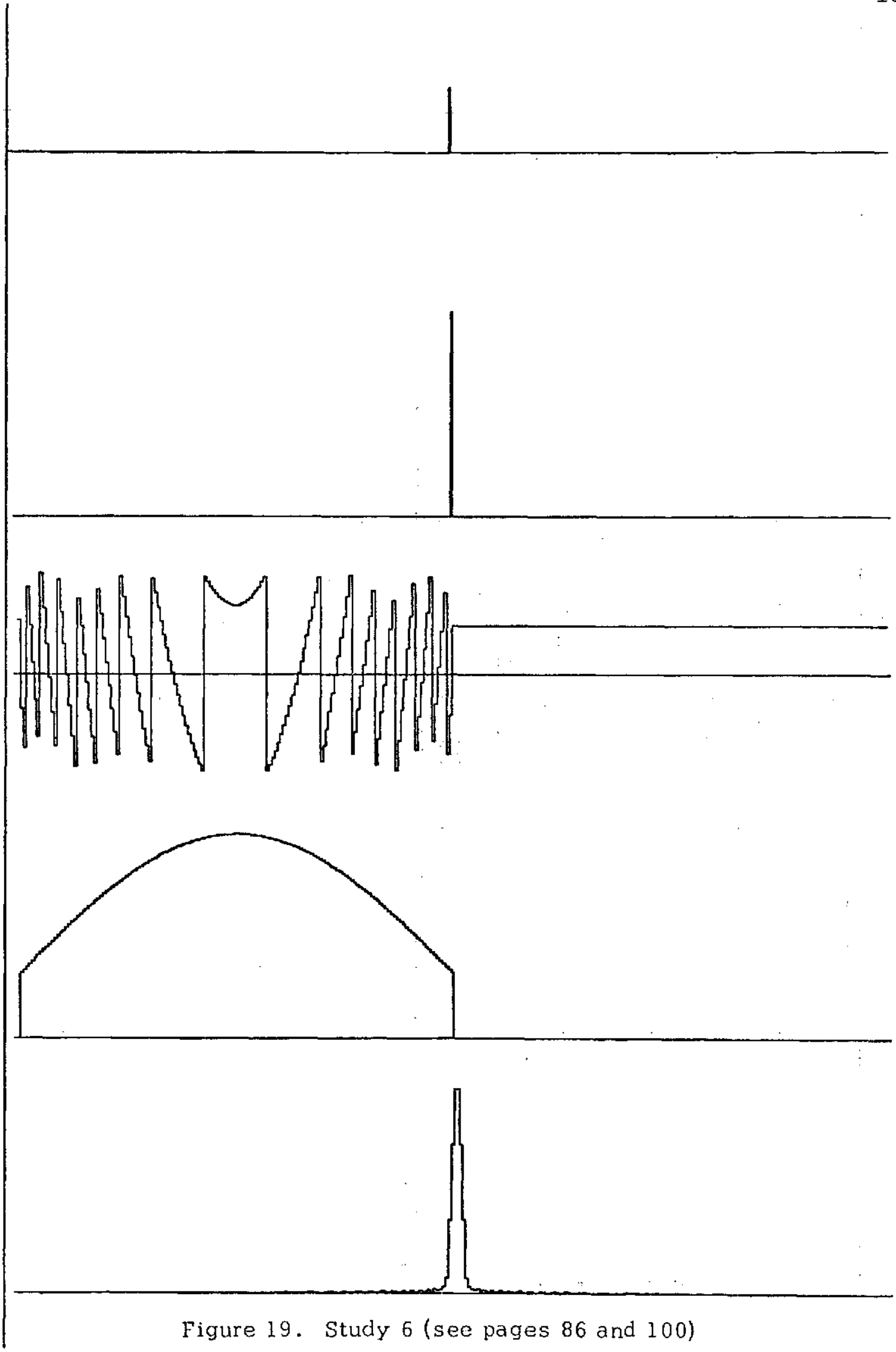


Figure 19. Study 6 (see pages 86 and 100)

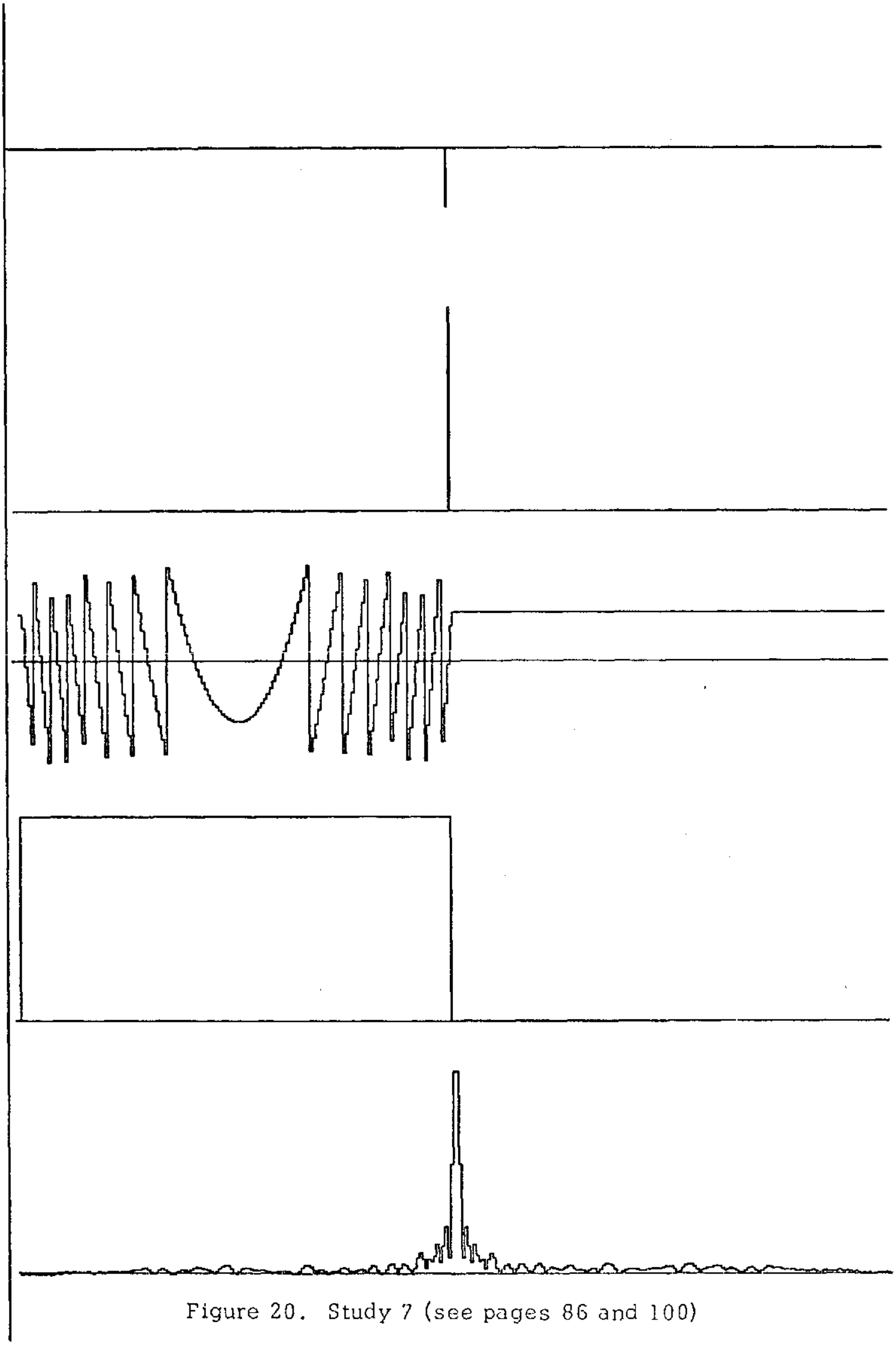


Figure 20. Study 7 (see pages 86 and 100)

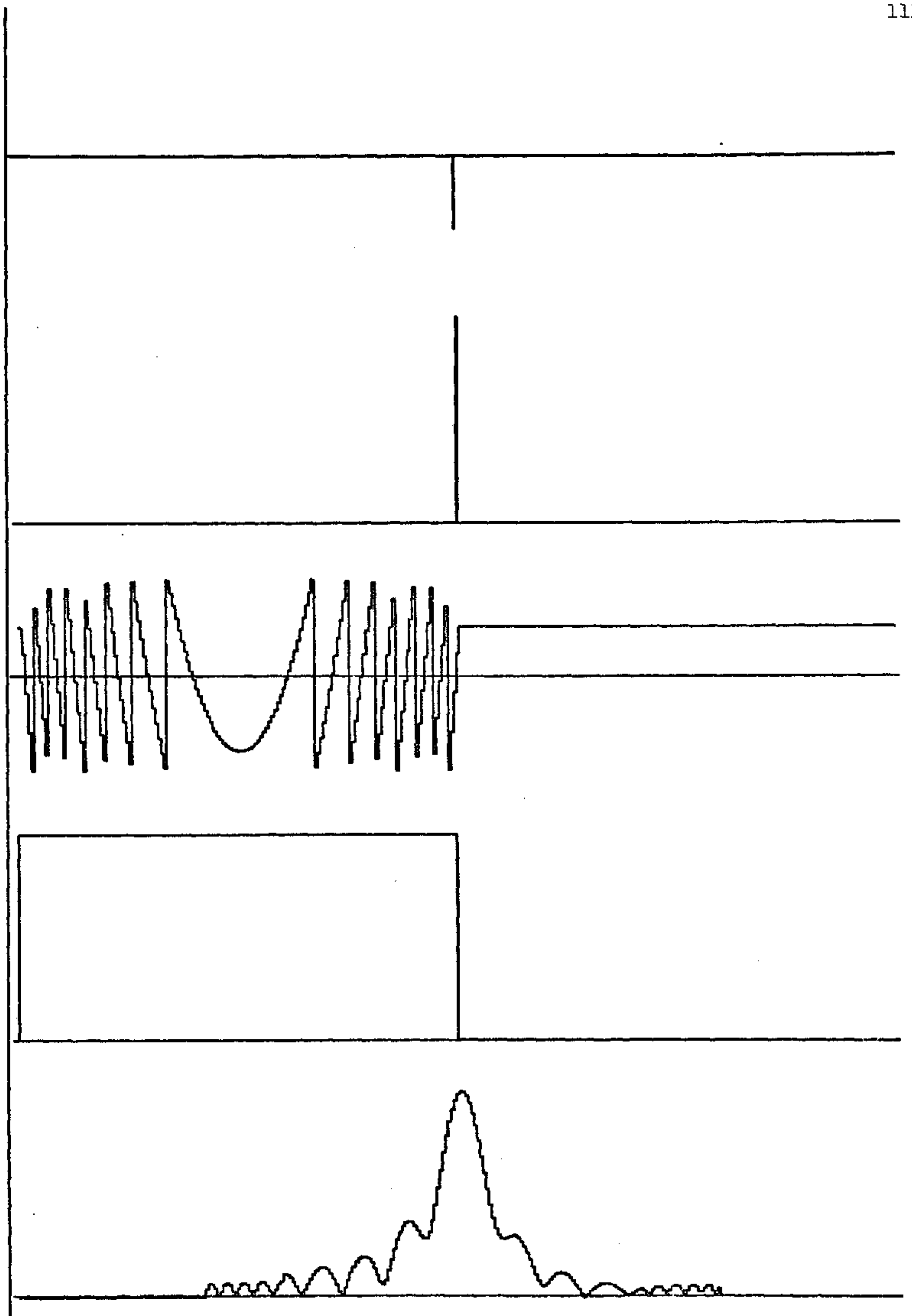


Figure 21. Study 8 (see pages 86 and 101)

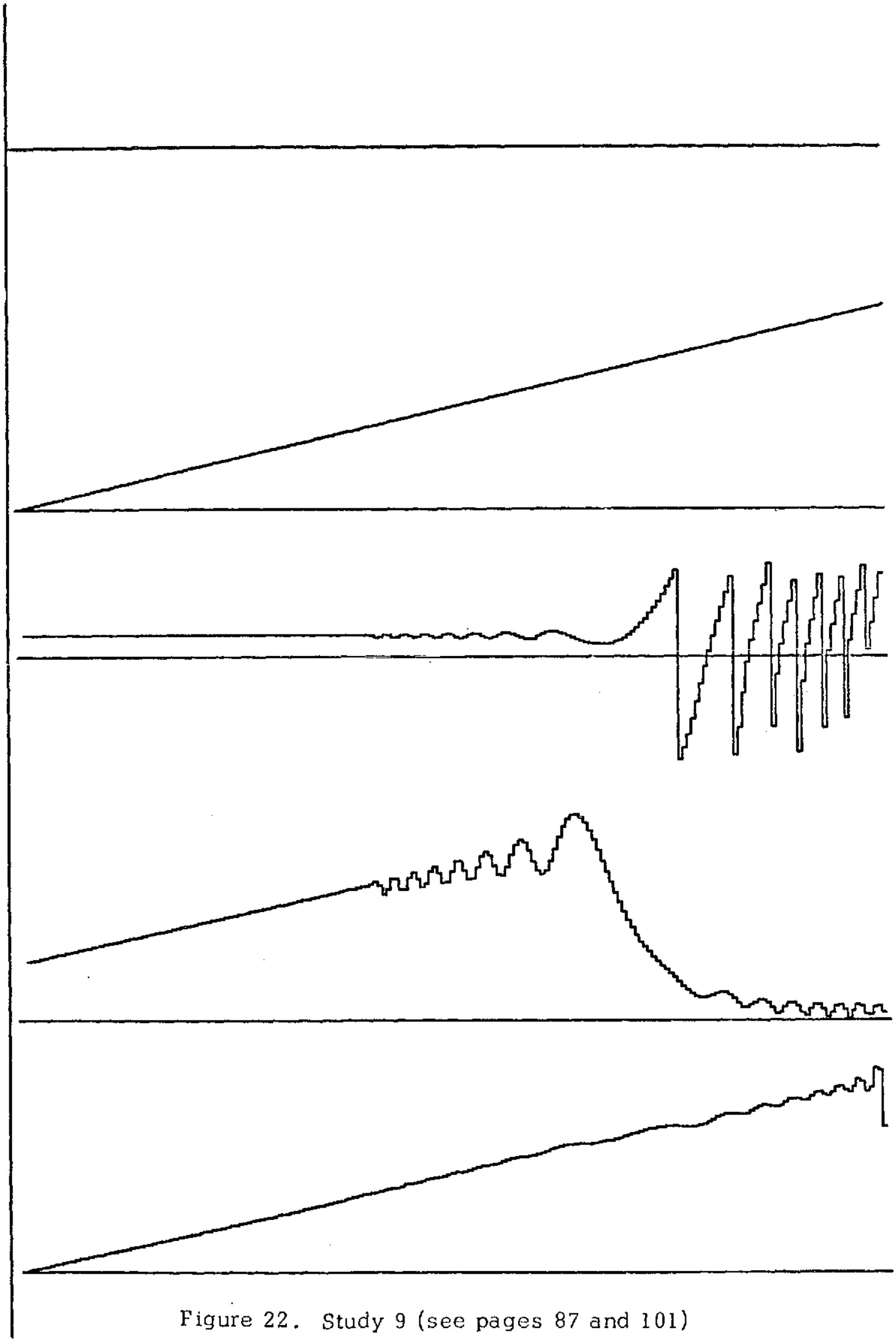


Figure 22. Study 9 (see pages 87 and 101)

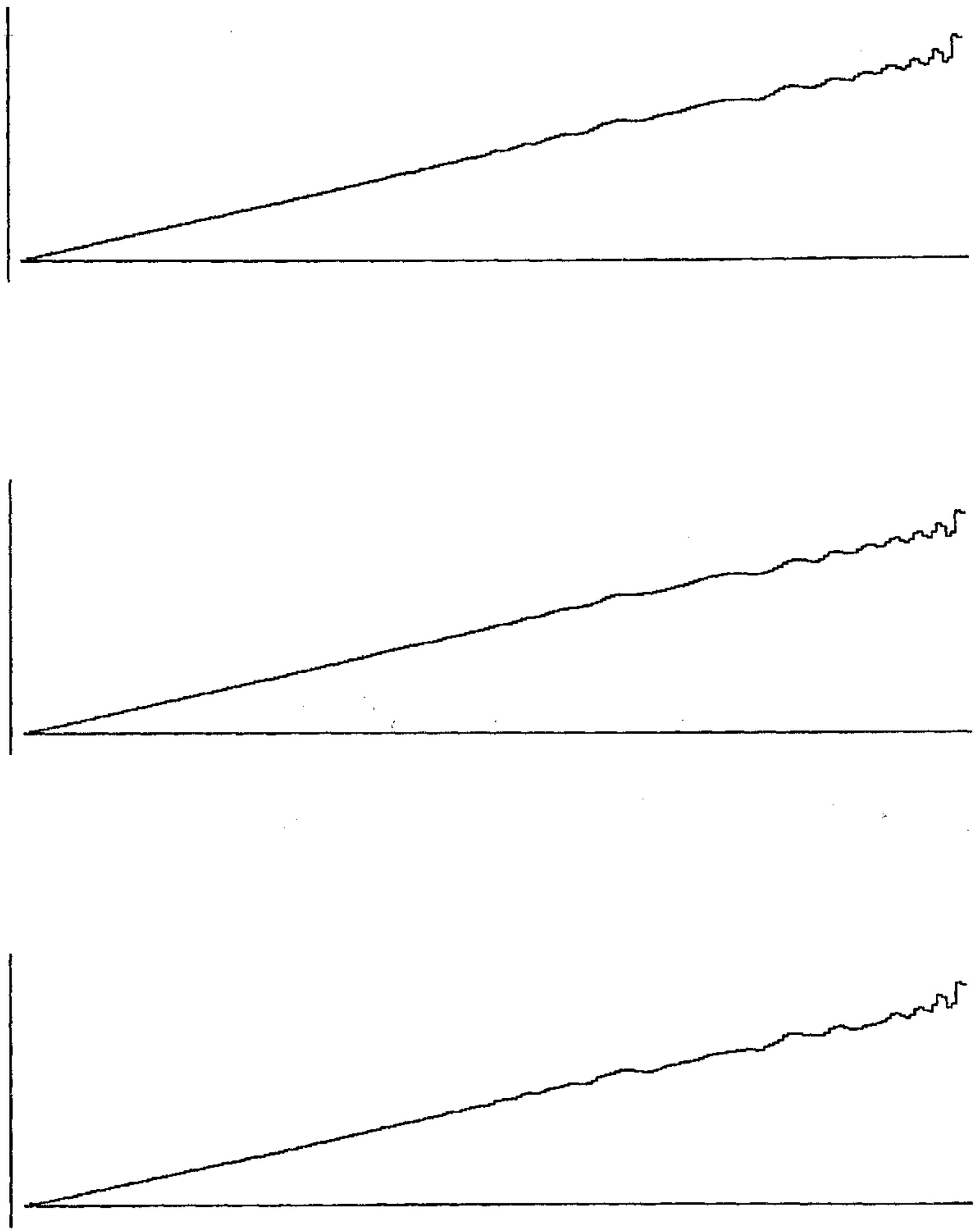


Figure 22a. Study 9 (see pages 87 and 101)

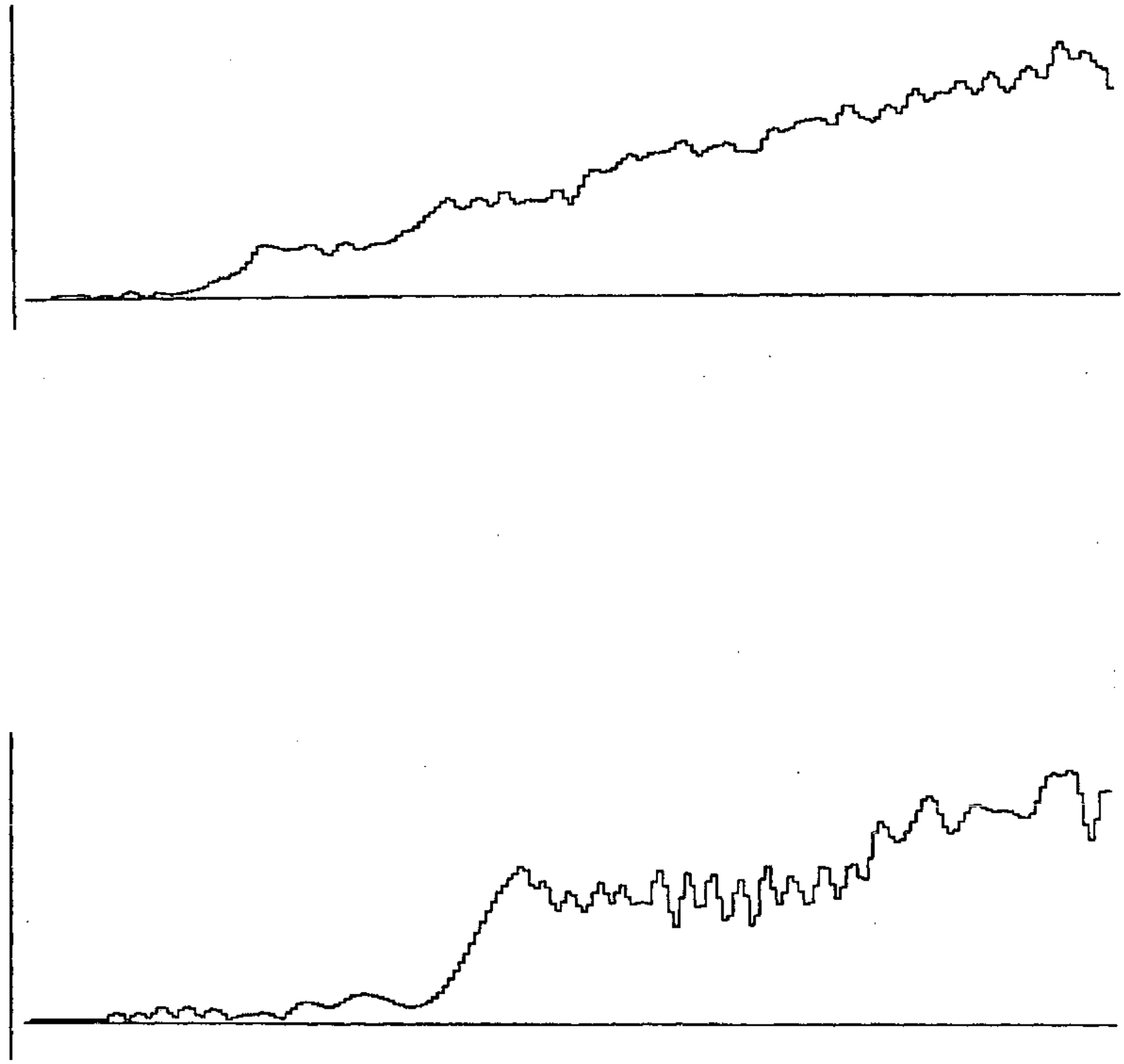


Figure 22b. Study 9 (see pages 87 and 101)

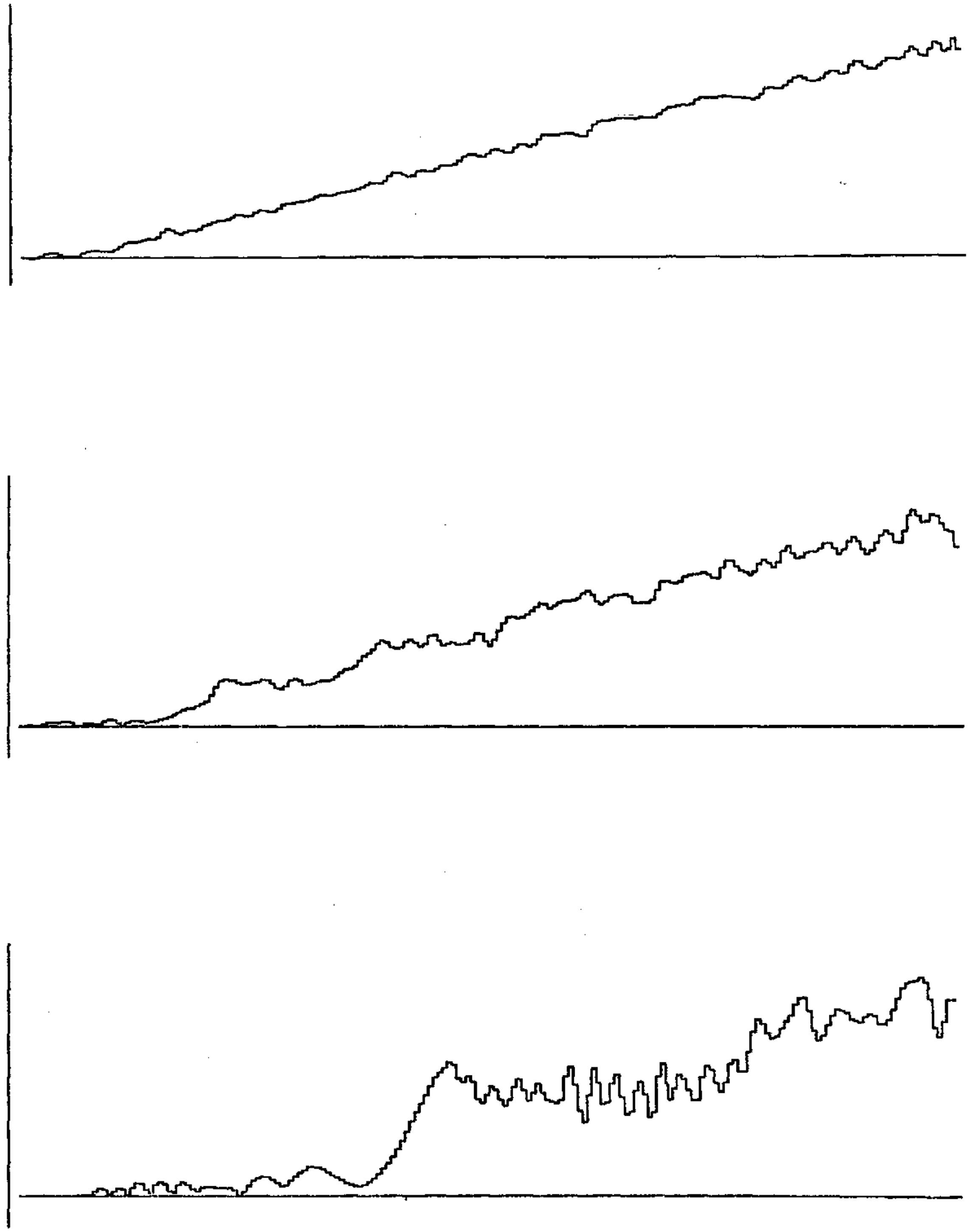


Figure 22c. Study 9 (see pages 87 and 101)

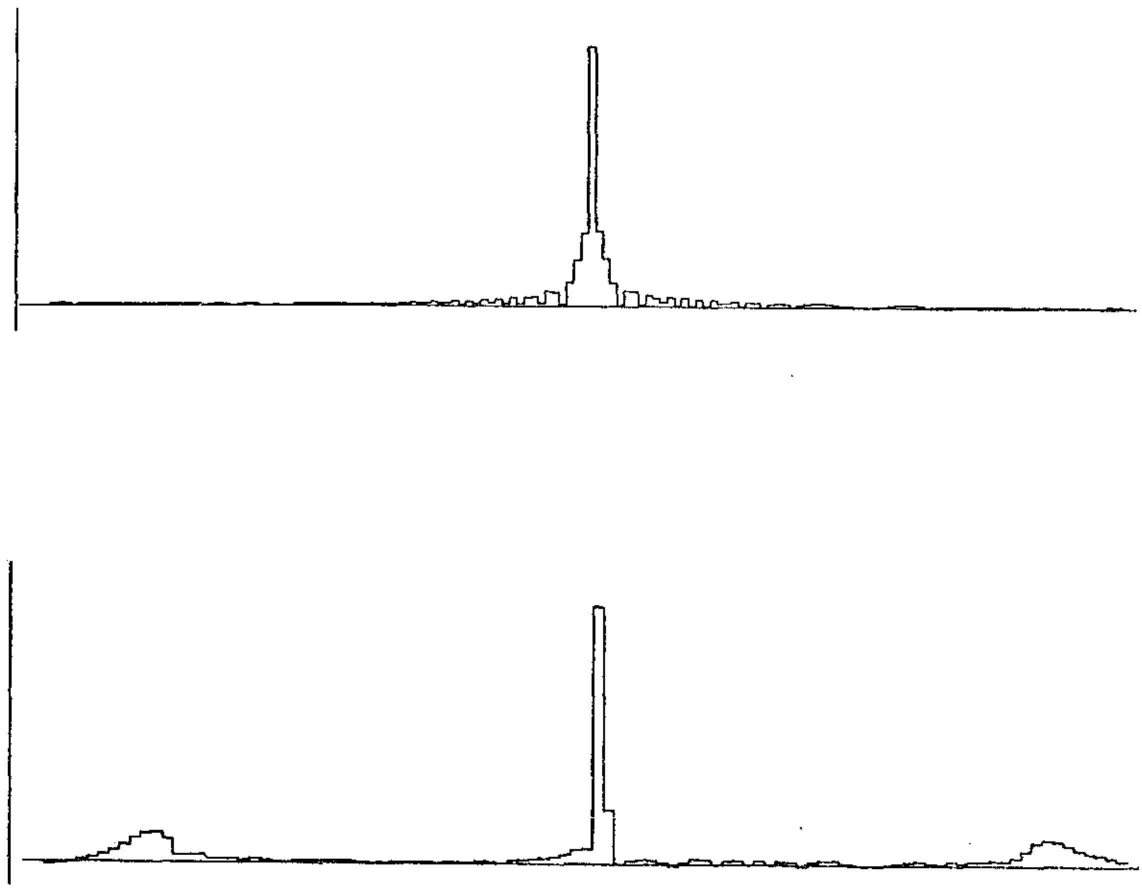


Figure 23. Study 10 (see pages 88 and 101)

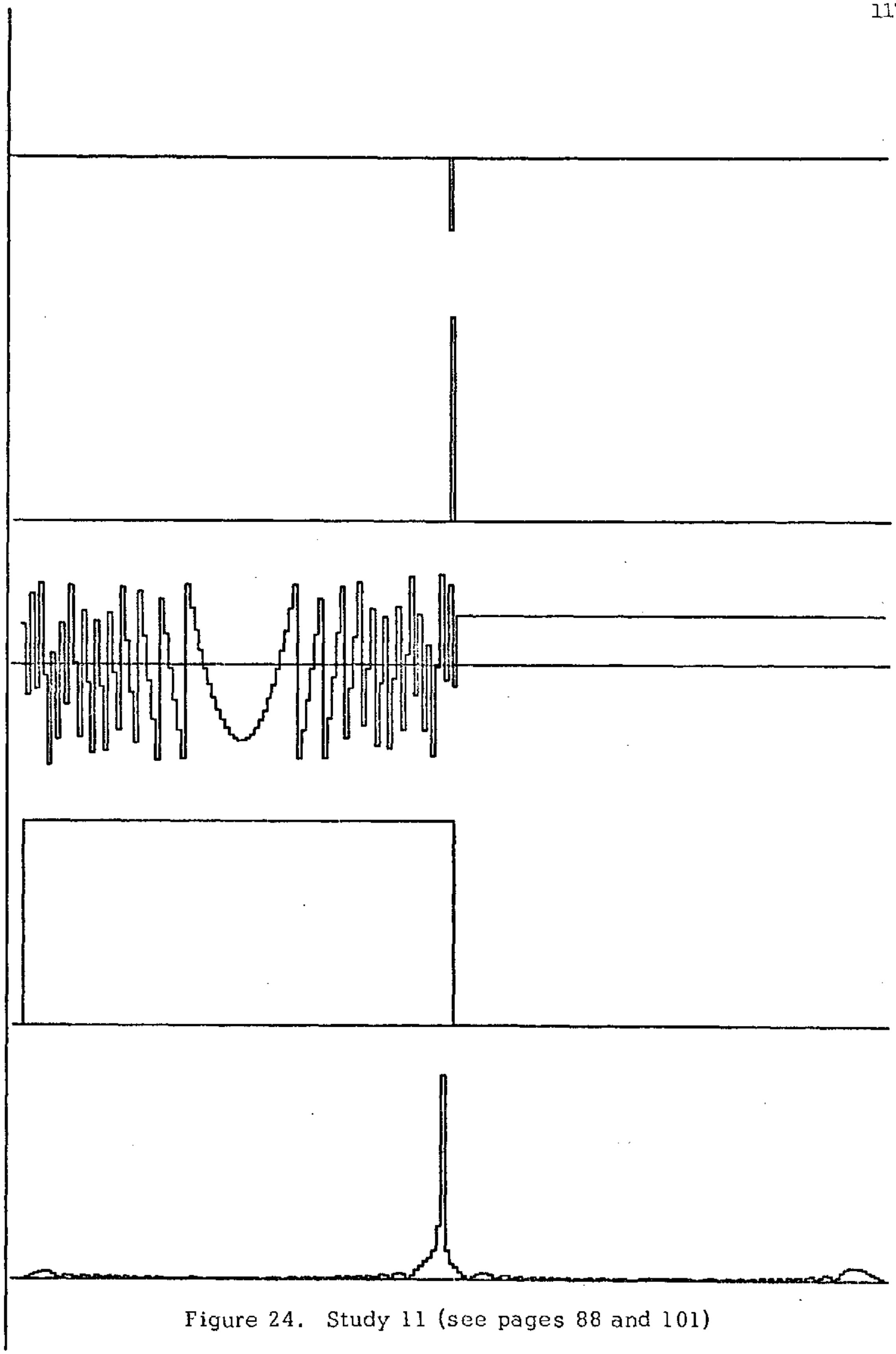


Figure 24. Study 11 (see pages 88 and 101)

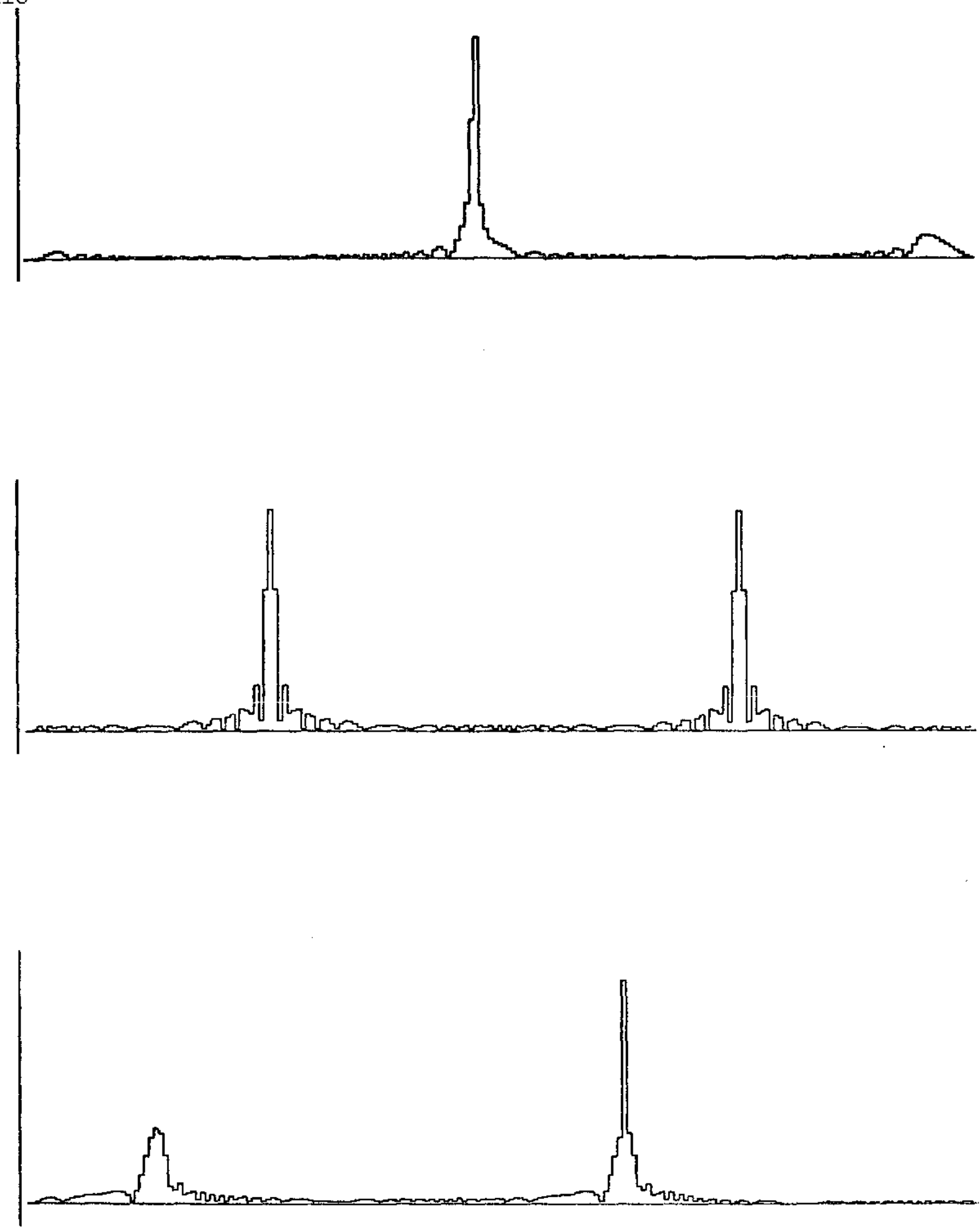


Figure 24a. Study 11 (see pages 88 and 101)

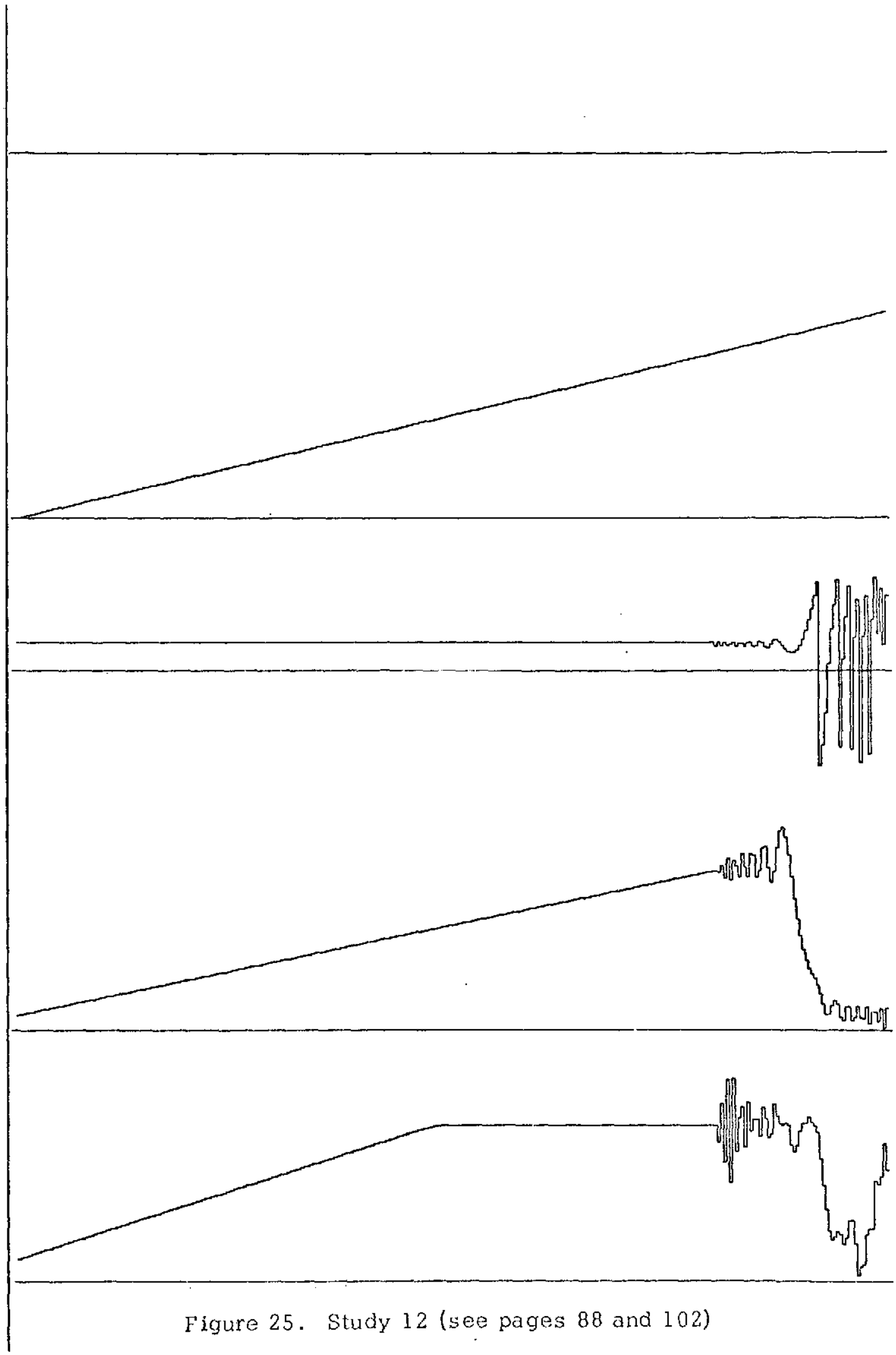


Figure 25. Study 12 (see pages 88 and 102)

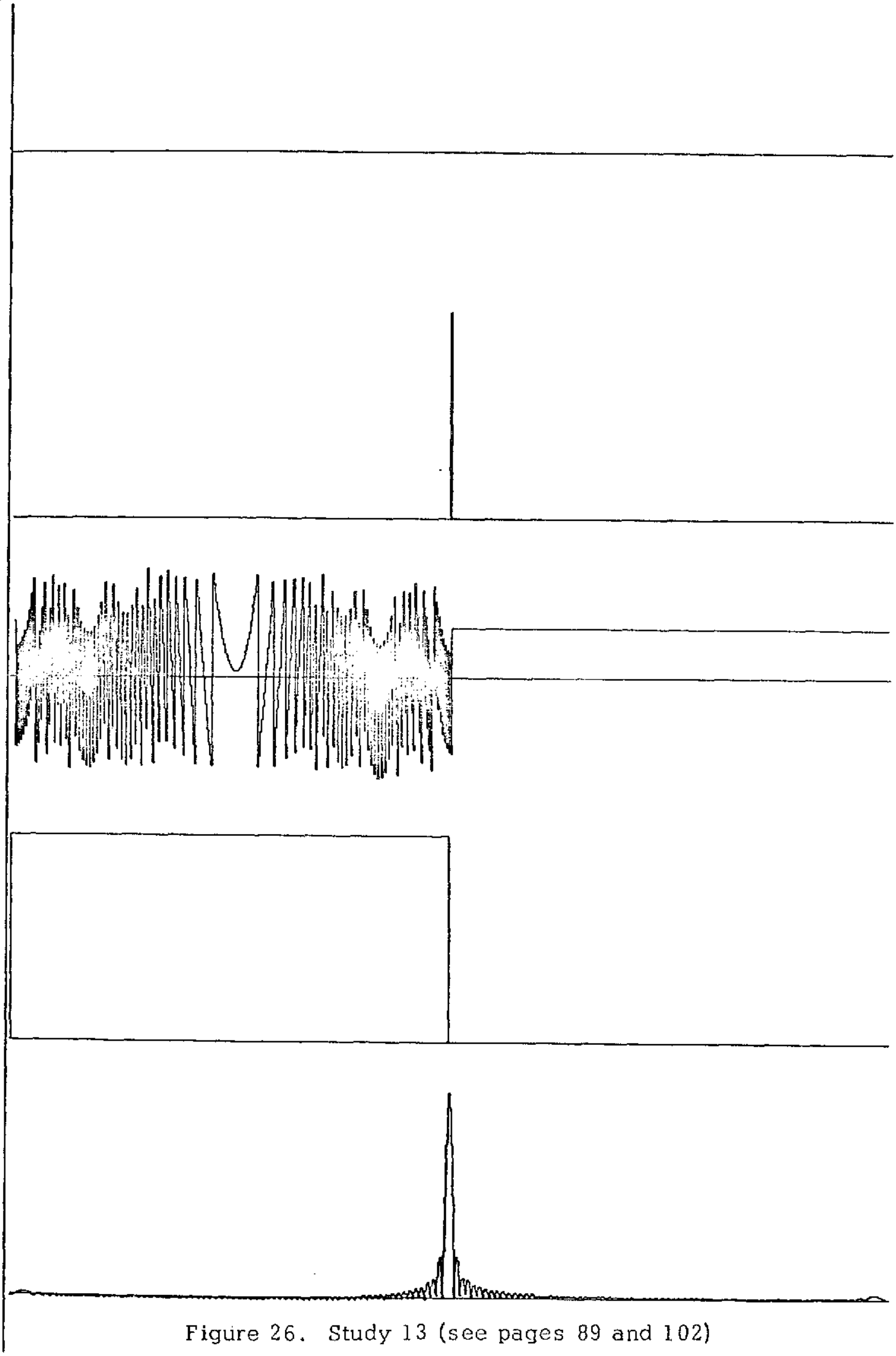


Figure 26. Study 13 (see pages 89 and 102)

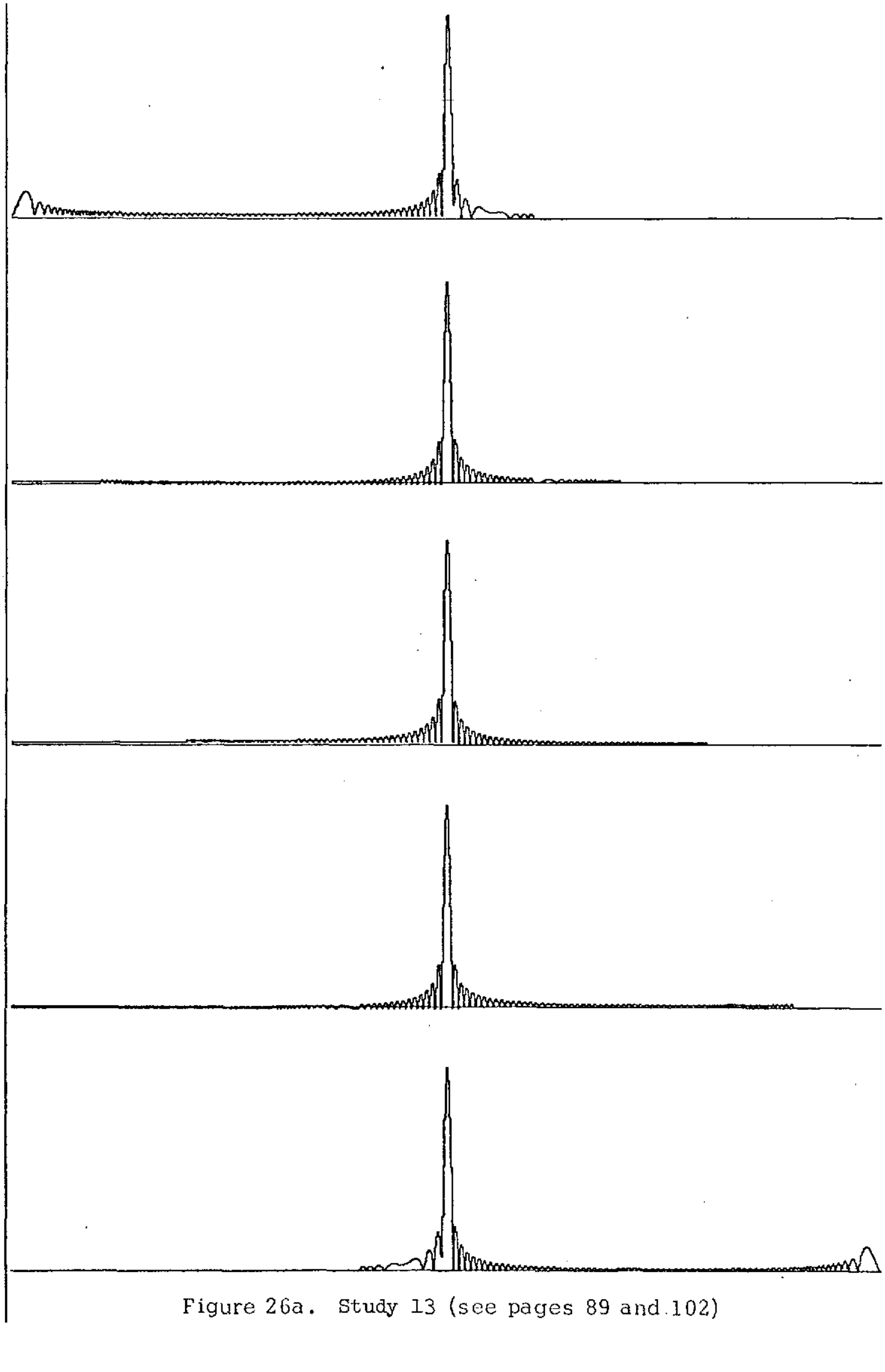


Figure 26a. Study 13 (see pages 89 and 102)

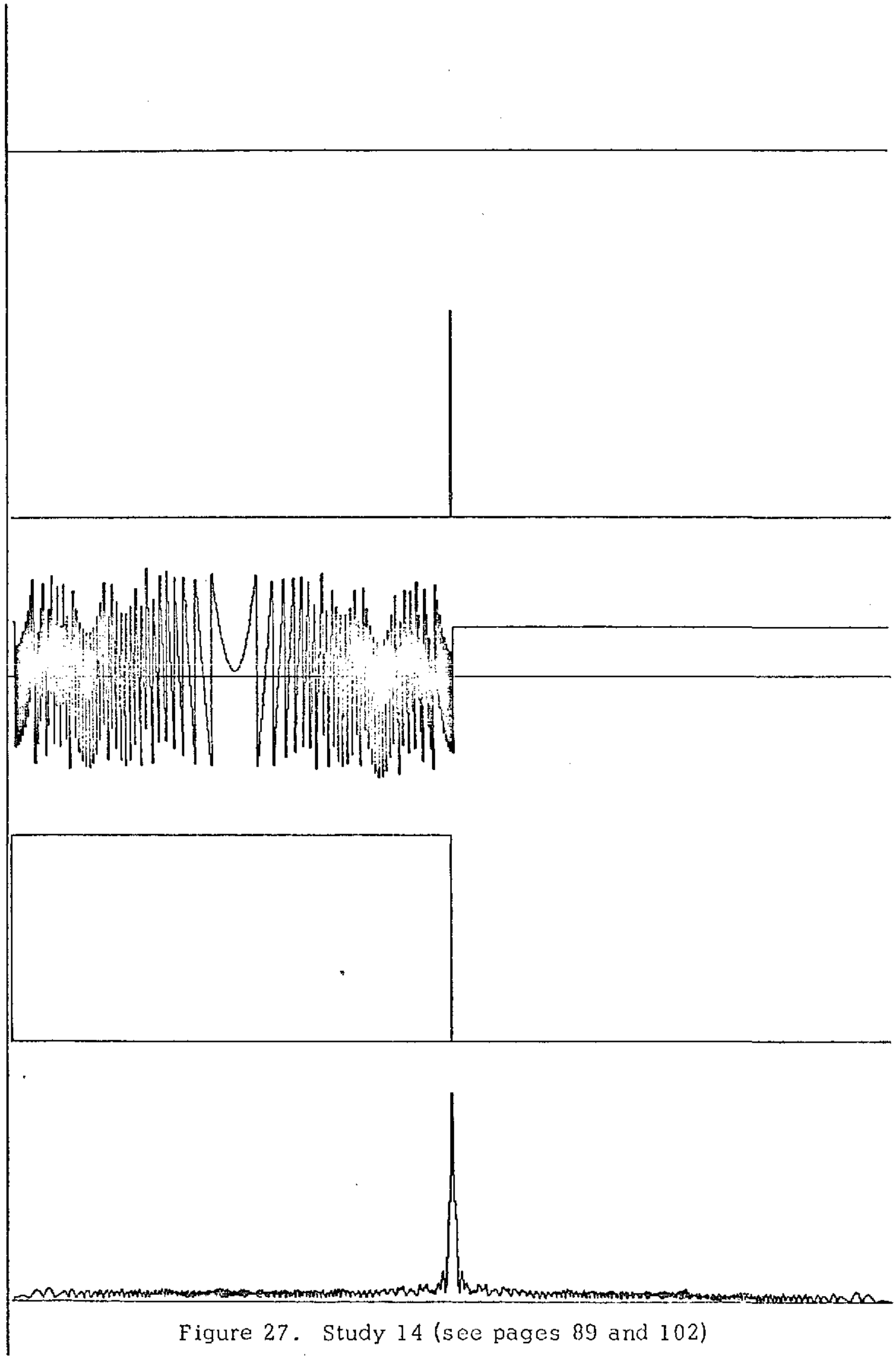


Figure 27. Study 14 (see pages 89 and 102)

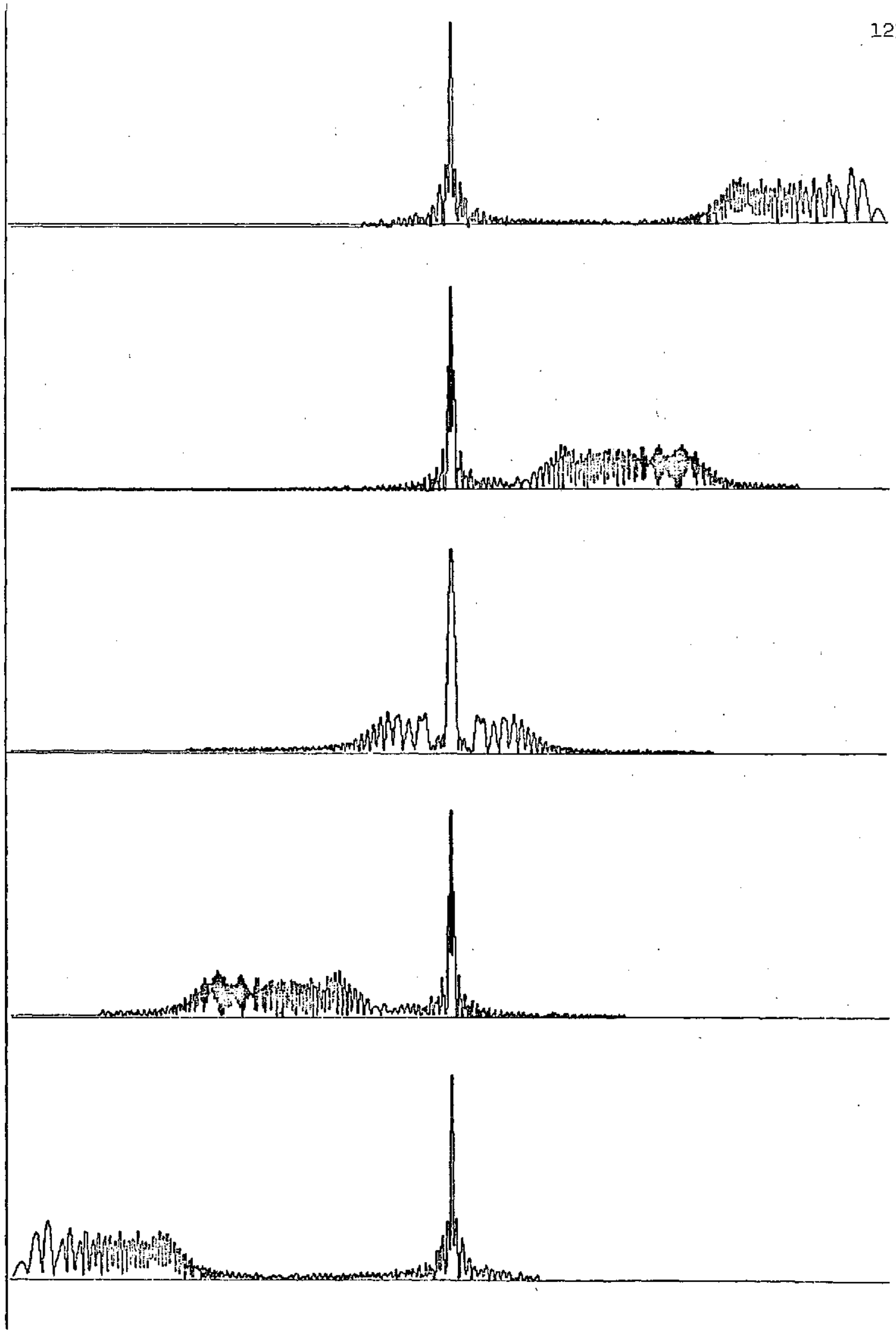


Figure 27a. Study 14 (see pages 89 and 102)

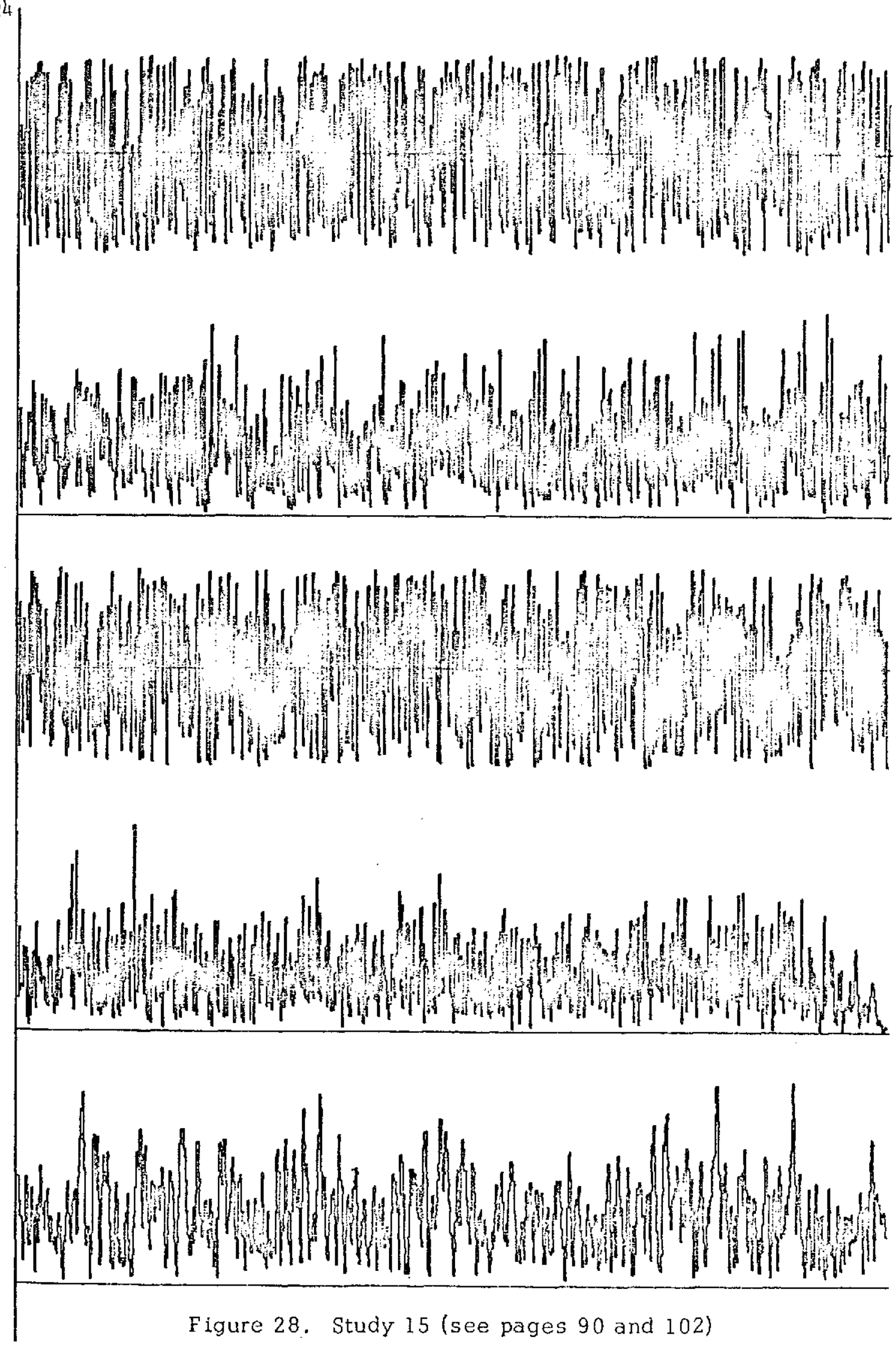


Figure 28. Study 15 (see pages 90 and 102)

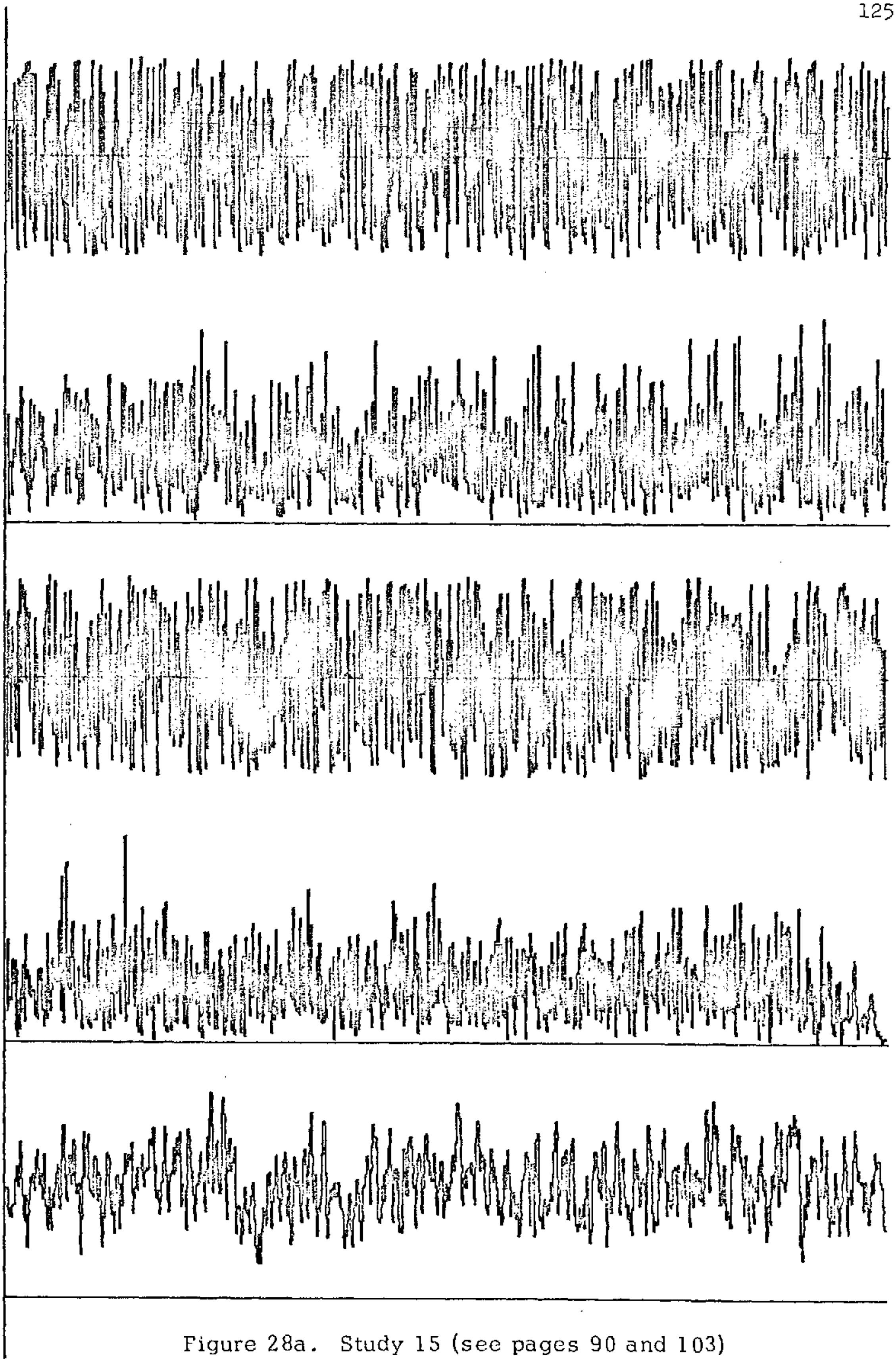
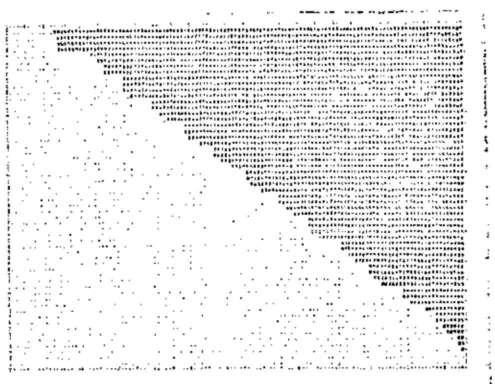


Figure 28a. Study 15 (see pages 90 and 103)

(1)

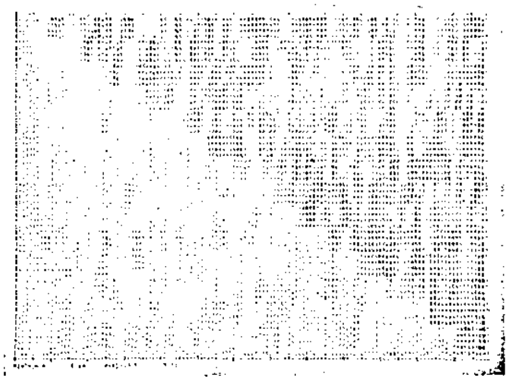
(2)



(3)

→
Flight direction

Figure 29 Study 17. Fully Focused synthetic aperture imaging over a boundary separating two fields whose non-statistically distributed scattering cross section differ by 9.54 db. (1) "True" map of fields, (2) SAR image with dynamic range of display matched to image dynamic range, (3) SAR image with picture dynamic range much smaller than image range.



(1)

(2)

(3)

→
Flight direction

Figure 29a Study 17. (1) "True" map of Rayleigh distributed scattering cross sections for 2 fields whose differential scattering cross sections differ by 9.54 db — M/STD $\approx \sqrt{3.6}$; (2) non-quadrature subaperture processing - 5 subapertures — M/STD $\approx \sqrt{9.0}$; (3) quadrature subaperture processing - 5 subapertures — M/STD $\approx \sqrt{18.0}$.

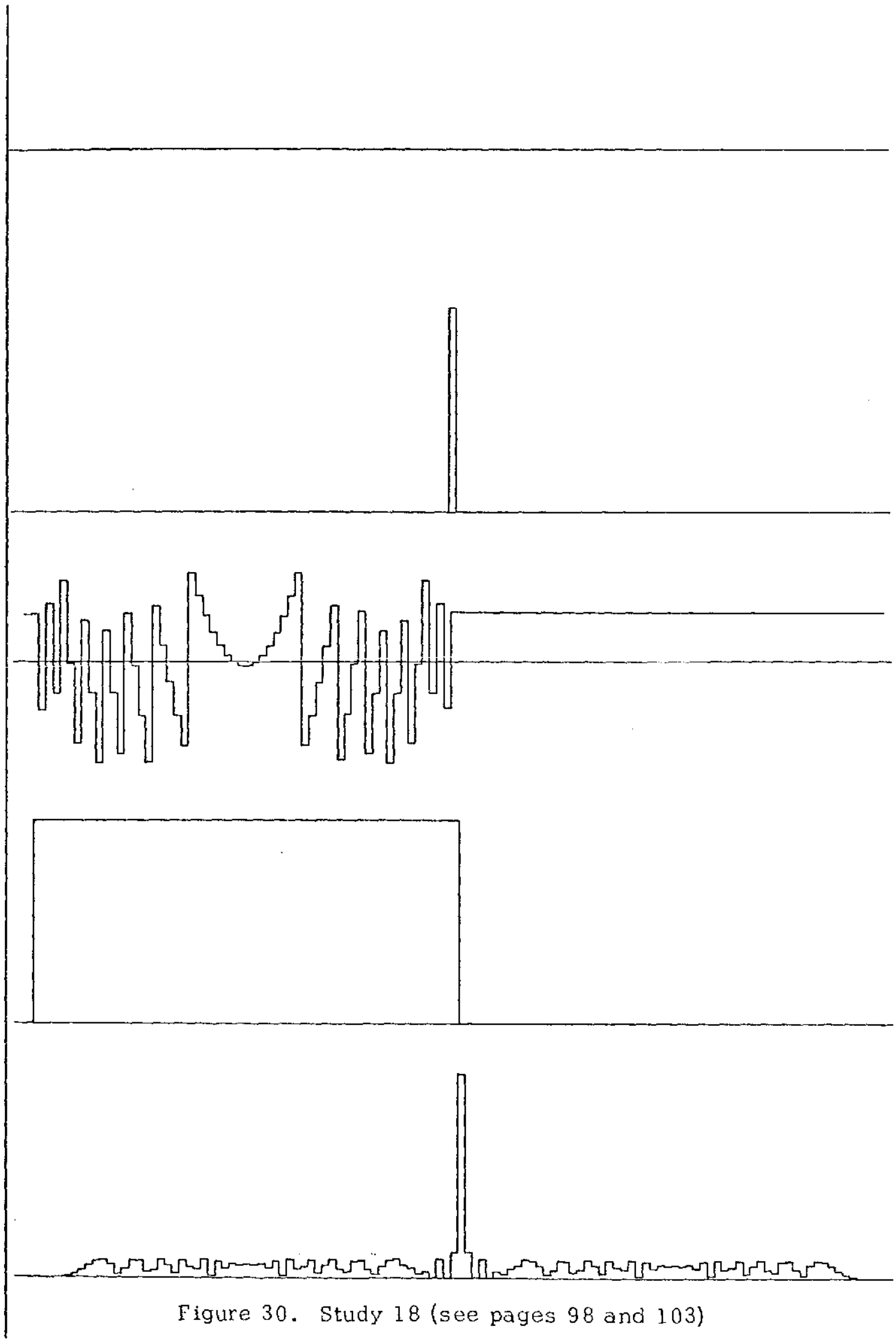


Figure 30. Study 18 (see pages 98 and 103)

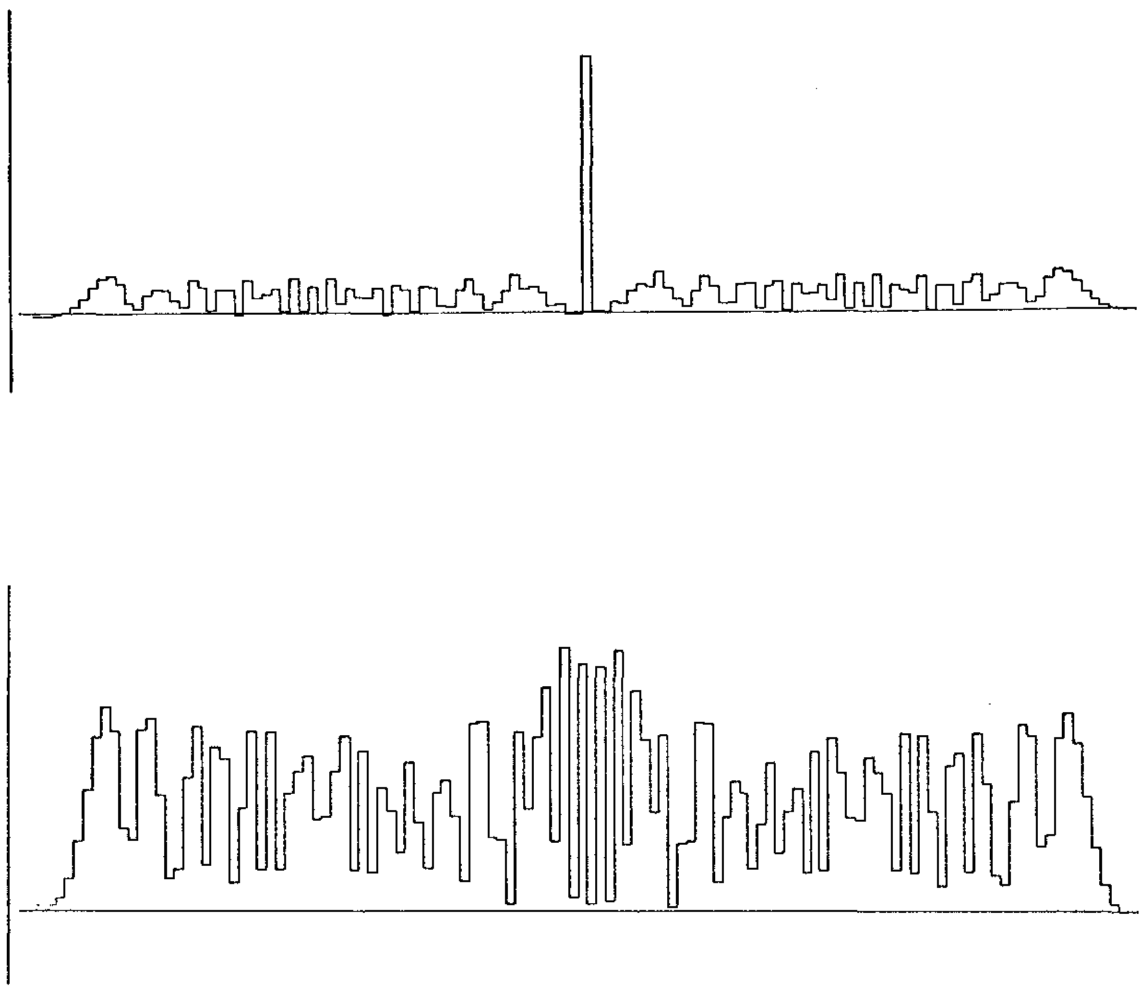


Figure 30a. Study 18 (see pages 98 and 103)

STORAGE AND SPEED REQUIREMENTS OF A SAR DIGITAL PROCESSOR

Chapter Synopsis

In this chapter the requirements are studied for a digital processor to achieve non-quadrature sub-aperture processing, as described in Chapter III. Parenthetically one notes that full focused processing with non-quadrature detection falls within this processing type (1 sub-aperture) and hence the requirements derived may be used as a comparison of full focused processing against multiple sub-aperture processing.

Table V is presented for easy determination of the processor requirements. The reader enters the Table with the number of bits to be used per number in the "working stores" (correlator section) and the minimum M/STD ratio to be characteristic of the radar imagery. The physical geometry of the imagery system provides the doppler bandwidth (B) of the return signal and the time (T) during which point target on the ground is in the nominal beam of the physical antenna. The best along-track resolution possible is v/B where v is the radar speed. The resolution degradation factor is (N) so that the computations are made for an along-track resolution of v/(BN). Computations according to Table V provide information on the trades available between processor size and speed, image resolution and image M/STD ratio.

The latter half of the chapter is devoted to a discussion of the present capabilities of Large Scale Integration (LSI) technology with regard to the achievement a digital processor. It is shown that standard commercially available circuits have the potential for realizing the circuits but the power dissipation of the processor is a major problem. On the other hand laboratory-demonstrated circuits employing "complementary metallic oxide silicon" (CMOS) techniques potentially hold the solution to the power problem. Based on the estimated state of LSI technology in the year 1975, the design of an SAR system employing real time digital processing is given.

Storage Requirements of SAR Digital Processor

It has been shown that one may trade along-track resolution in a radar image, for enhanced image mean to standard deviation ratio. This in turn may be traded to ease processor storage and cycle time requirements. The calculations that follow will be based entirely on a single channel (non-quadrature detection) receiver. Compared to quadrature processing, the number of operations that must be performed in the processor with single channel reception is reduced by 75 per cent and the processor storage requirement is halved. The price for this is that image mean to standard deviation ratio is reduced by 3 db. However, along-track resolution remains unchanged and the subaperture processing scheme is virtually unaffected by non-quadrature processing except for the loss of 3 db in M/STD.

Two principal levels of storage for each range bin as shown in Figure 31, are envisioned in the final processor. The first level is called the working store. It is a series of L bit words containing the amplitudes of the last J pulses. It is against this series of numbers that the correlation process is performed. Each of the J numbers is multiplied by a corresponding number according to a reference function supplied by a read only memory. The J products are accumulated in an adder and the magnitude of the final adder result in transferred to one of K cumulative adders which form the second principal storage level - the averaging store. Once an averaging store has received P numbers for cumulative addition, the sum is transferred out of the processor to a display or a permanent storage device and the store is cleared. The last number represents a grey level sample point on the final radar image.

The only scheme that will be investigated in arriving at storage requirements will be that of employing sub-apertures for generating uncorrelated "views" of a given terrain area (see Chapter III).

Assume that the time-bandwidth product of the return signal from a point target at the side-looking radar is TB. Under matched filtering the maximum number of resolution cells in the along-track direction of the physical antenna beam on the ground is TB and the length of each resolution cell in time is B^{-1}. Using the subaperture scheme let N non-overlapping contiguous sub-apertures be created. This makes it possible to degrade the along-track resolution and to enhance the image M/STD ratio by the factor $\sqrt{N}$. At the same time the working store need be only $J = TB/N$ words long. (For matched filtering with a single channel it would have been TB words long.) Because the along-track image bandwidth is $B/(2N)$ a point on the image line must be measured every N/B seconds. Thus the first sub-aperture creates a line of TB/N^2 (i.e., the TB product of the sub-aperture) samples before the next sub-aperture contributes to the averaging. Hence, the total number of words required in the averaging store is $K = TB/N$. However, the size of word in bits in this store is considerably larger than that of the words in the working store (L bits long).

Starting from a knowledge that return values from the IF strip may be quantized to L bit words and assuming the same quantization (L-bits) for the reference function words, the product of two L bit words may be a 2L bit number which will be accumulated in the cumulative summer (see Figure 31). The number of returns collected in the summer before being added to an averaging store is TB/N. Therefore, the cumulative summer needs to be $[2L + \log_2 (TB/N)]$ bits long. On the other hand for the averaging store, a conservative design requires a word to be $[2L + \log_2 TB]$ bits long inasmuch as there are N additions to an averaging store from the cumulative summer. The results of this analysis are summarized in Table V.

L BIT QUANTIZED RETURN

↓

BUFFER STORE → WORKING OR CORRELATION (FAST) STORES

↓

REFERENCE FUNCTION → CORRELATION TAKES PLACE AS A CUMULATIVE SUMMATION IN THE CUMULATIVE SUMMER

↓

ADDED TO NUMBER FROM AVERAGING STORE

TAKE LOGARITHM

TAKE ANTILOG

AVERAGING (SLOW) STORE

↓

DISPLAY

FIGURE 31 FLOW DIAGRAM OF PROCESSOR FOR EACH RANGE BIN

For a realistic example consider the space radar applications example outlined in Chapter I. The time (T) for the signal is 1.377 seconds. The doppler bandwidth B is 3500 Hz; the time bandwidth product (TB) is 4820. It is desired to reduce the resolution in the along-track direction by a factor (N) of 15. At the same time, the M/STD ratio will be enhanced by a factor of $\sqrt{15}$ in the image. Based on simulation studies (see Figures 22 and 22C) it appears that a working store word of L= 5 bits is good enough. The working store needs to be 321 words long at 5 bits per word. However, the averaging store requires 321 at 23 bits per word. Thus, the total storage per range bin (the sum of working: 1605 and averaging: 7383, store bits) is 8988 bits. Since there are 1334 range bins in the space application example, the total principal storage is 12.0 megabits. This represents an impractically large storage for a small spacecraft.

It is noted that word lengths in the averaging stores, for complete accuracy, are very long (of the order of twenty bits). Reducing the size of these stores involves the addition of quantization noise in the final image. To judge the magnitude of the error involved in such a storage reduction the following conditions are assumed. The range of differential scattering cross sections in the image is taken to be 40 db. Since the targets are modelled as being Rayleigh distributed, the range of quantization levels will be for the largest scattering cross section plus one standard deviation to the smallest scattering cross secion. The Rayleigh power distribution is chi-squared two and since the standard deviation is equal to the mean, the range of powers is from 1 to 20000 and the range of voltages is from 1 to 150. The averaging stores contain voltage amplitudes and 20 bit stores are larger than the job merits.

Because the mean-to-standard-deviation of each addition to an averaging store is the same, it is natural to quantize in logarithmic increments. The number in the cumulative summer immediately before addition to the averaging store will have a dynamic range of 150:1.

The n^{th} quantum level will be expressed an δ^n where $\ln \delta$ is $(\ln 150)/2^B$. B is the number of bits required for the final number in the cumulative summer (though in fact the summer is much larger than this).

The ratio of the mean square error, due to requantization of the final number in the cumulative summer, to the mean of the final value squared will be according to Schwartz[8]:

$$\frac{(\delta^{n+1} - \delta^{n})^2}{12\,\delta^{2n}} = \frac{1}{12}(\delta - 1)^2$$

But for Rayleigh targets the variance to the square of the mean is 0.275. Hence, with requantization this ratio becomes:

$$0.275 + \frac{1}{12}(\delta - 1)^2$$

If there are N uncorrelated numbers to be added to an averaging store, the final M/STD ratio will be:

$$\sqrt{\frac{12\,N}{3.3 + (\delta - 1)^2}}$$

Returning to the space application example and employing the full number of sub-apertures (15), the size of the word in the averaging stores need be no greater than 9 bits (5 bits for requantization plus $\log_2(15)$ for the 15 subapertures) to achieve an image M/STD degradation no greater than 0.1 db. because of the requantization. With requantization the total principal storage in the processor becomes 6.0 megabits. At this point one may decide to surrender some M/STD ratio or image swath width for a lesser storage requirement.

If the system and quantization noise and the clutter interference were negligibly small, the radar image would still appear to be noisy. Assuming that the ground target voltage reflection coefficient amplitudes are Rayleigh distributed for a population of homogeneous targets and that the phase of the voltage reflection coefficient is uniformly distributed between π and $-\pi$, the ratio of the mean voltage to the square root of the voltage variance from a resolution cell (the inherent

M/STD ratio for a homogeneous class of targets) is 5.61 db. Without averaging to enhance the M/STD ratio, this number is the determining factor in gray level resolution in the image. This is the best signal to noise ratio SAR imagery even if the system is operating fully focused for its best close target resolution. However, because the system that is being considered here is not quadrature detected, and there is no offset frequency, this number is reduced by 3.0 db (see Table IV page 93). Figure 29a on page 127 gives some idea of the effect of the M/STD ratio in an image. Suppose that the imagery is specified to have 30 meter resolution and a minimum M/STD of $\sqrt{18}$. This could be achieved under subaperture processing for the space application example by using only ten of the possible 15 non-overlapping subapertures. Accordingly the working store would contain the same number of bits as before but a saving would be affected in the averaging store. If G is the total number of non-overlapping sub-apertures used when N is the total number possible, the total number of words in the averaging store (K) is given by:

$$K = \frac{TB}{N^2} G$$

The number of bits per word in the averaging store is determined on the basis of the M/STD degradation allowable because of requantization error (P in db down) and the range of voltages expected as the final number in the cumulative summer (R_v). In the example, for ten sub-apertures, the averaging stores per range bin contain 214 words each 9 bits long. Hence, the total principal stores for 1334 range bins contain 4.75 megabits. The General Electric 635 general purpose computer has a main frame storage unit of some 7 megabits.

It is interesting to note that the speed of the processor does not appear to be a determining factor in the realization of a digital processor as Harger[6] intimates it might be. Because the image lines generated for each range bin are independent of each other, there appears to be no reason why they may not all be processed in parallel. Under such processor organization the cycle time required in the "working stores" is $N^2/(B^2TG)$ per word or 1.3 micro-seconds per word for the space application example. This is also the time permitted to perform a multiplication. The averaging stores are accessed only one $(TB/N)^{th}$ as often in each range bin leading to a cycle time of N/BG. For the space example the averaging store cycle time per word is .43 milliseconds. The large difference in cycle time between the averaging and working stores suggests that a disc or a drum with large capacity and relatively slow access might serve as the averaging store.

In view of the fact that the digital processor storage is so large under sub-aperture processing, a comparison of fully focused SAR imaging employing non-quadrature detection and along-track resolution degradation is in order. Such processing would be done for the finest along-track resolution possible and then a series of N image points would be summed to degrade resolution by the factor N. In the example situation the working store per range bin requires TBL bits and the averaging store size is negligable. Each range bin would have 24,000 bits and the total principal storage in bits would be 32 million bits compared 6.0 megabits for sub-aperture processing to the same M/STD ratio. The sub-aperture scheme offers the user the option of taking some loss in M/STD to reduce the storage requirement still further. This is not an option for full focused processing with sequential averaging. Furthermore, storage cycle times are prohibitively small with full focused processing.

One notes that the basis for the word access times calculated according to Table V is a single range bin requirements. That is it is assumed that the processing organization will be such that each range bin will be processed in parallel with all the others. This is in line with Khambata's[11] idea of a multiprocessor with large scale integrated circuits. He writes:

> "If an entire data processor could be built on a small number of chips (ideally on a single chip), why not connect several such processors to a common bank of memories? Such a multiprocessor would have many

> advantages that could not be realized by our conventionally organized systems... In such a system, the speed demands on each individual processor need not be too great. In fact, each processor could be substantially slower than currently used serial type machines... In our sequential or serial type machines speed has been a serious consideration. Since computations and operations take place in a serial fashion, improvements in speed have been achieved only by the brute force method, that is by improving the speeds of the circuits and sub-system. An upper limit is fast approaching, however, and at the present time transmission line problems, rather than circuits speeds, are the limiting factors in systems speeds. Thus we find ourselves using systems that incorporate very high speed switching circuits which nontheless remain idle most of the time because of the way the system is organized. Obviously this is an inefficient use of high-speed circuits."

The question of whether such a digital processor is realizable with the present circuit technology available remains. The answer to this question is a very highly qualified yes. It has been shown for of the space application example that a 4.75 megabit memory is required with a word access time of 1.3 microseconds. According to the Texas Instruments, one 16 bit active-element memory chip #SN5481 has a typical access time of 20 nanoseconds and a typical average power dissipation of 275 milliwatts. Obviously, the access time is more than adequate but the power dissipation would present a problem. For an 4.75 megabit memory this represents a power of 82 kilowatts. Undoubtedly the power dissipation is one reason why LSI memories have yet to make their appearance in main frame memories of large general purpose computers. However, Texas Instruments is currently working on the development of a one square inch chip to store 4000 fully addressed bits (the decoding network will be on the chip).[22] This chip will dissipate 1.6 watts and will have an access time of half a microsecond. The total power for a 4.75 megabit memory would be 1.9 kilowatts. However this chip is not scheduled for marketing until late 1971.

TABLE V

Summary of Cycle Times and Storage Capacities for Sub-Aperture Processing

SYSTEM PARAMETERS:

Doppler Bandwidth of Return Pulse	B
Time for point target in physical antenna beam	T
Radar system velocity	V_o
Finest Along-Track Resolution	V_o/B
Along-Track Resolution Degradation Factor	N
Image M/STD improvement factor due to processor averaging	$\sqrt{G}$
The largest number of non-overlapping sub-apetures possible in processing	N
Total number of subapertures employed in processing	G
Requantization degradation in M/STD	P (db, negative number)
The square root of the range of differential scattering cross sections to be imaged	R_v
Logarithmic quantum in the averaging stores	$\delta = 1+1.82\sqrt{e^{-.23P}-1}$
Requantization bits	$B = \log_2 \left[\frac{\ln R_v}{\ln \delta}\right]$ raised to the next highest integer
Averaging store counter in bits	$C = \log_2 G$ raised to the next highest integer
Number of bits per word in working stores	L
Number of words in working stores per range bin	TB/N
Number of words in averaging stores per range bin	TBG/N^2
The total number of bits in working stores per range bin	TBL/N

The total number of bits in averaging stores per range bin	$(B+C)TBG/N^2$
The number of bits required in the cumulative summer	$2L + \log_2(TB/N)$
Number of words required in the buffer stores per range bin	N
Number of bits in buffer storage per range bin	NL

<u>FOR SUB-APERTURE PROCESSING WITH SERIAL PROCESSING IN EACH RANGE BIN</u>:

Working store word maximum cycle time (same as multiplication time)	$N^2/(B^2TG)$
Averaging store word maximum cycle time	N/BG

Heat dissipation is a problem that designers of integrated circuits must solve before their circuits can be used in large numbers in any single mechanism. It is interesting to read in a paper by J. J. Suran the details of one special purpose processor being designed around integrated circuits. Suran gives the following breakdown on the number of elements to be employed in the computer and on the power each dissipates.

Component Count for Special-Purpose Data Processor

Component Level	Numbers	Types
Transistors	1,300,000	6
Gates	400,000	6
Chips	4,000	8
Module	500	8
Backboard	50	5

Power Dissipation for Special-Purpose Data Processor

Component	Power needed (watts)	Power density W/cm^2
Transistor	.0005	15.5
Gate	.0015	7.8
Chip	.170	3.1
Module	1.36	.15
Backboard	13.6	.0 5

The number of elements is quite large though still less by an order of magnitude than that required for the digital processor required in the space application example. The power dissipation problem is considerable especially because of the sensitivity of solid state active elements to changes in temperature. Nevertheless Suran feels that use of large numbers of active elements in small volumes is going to be realized in the decade of the seventies. For those who cannot believe this of the future, he is prompted to quote from the past. In 1946 John von Neumann said,[24] "We saw earlier that a fast memory of several thousands words is not at all unreasonable for an all purpose instrument. Hence about 10^5 flip-flops or analogous elements would be required! This would of course be entirely impractical." Today main frame memories in large computers have capacities of 7 to 10 megabits.

LSI memories, now, have access times of from 10 nanoseconds to 2000 nanoseconds.[25] However, Texas Instruments does not foresee large main frame memories constructed with LSI technology until sometime in 1974 or 1975. With this in mind one is forced to the conclusion that constructing a space SAR digital processor today with a 4.75 megabit LSI memory would not be economic.

Reports are current in the literature about a new version of LSI technology called CMOS for complementary Metallic Oxide semiconductor. Apparently this techology will be fast and require very little in the way of power dissipation. R. A. Stehlin[26] reports for example:

> "Integrated circuits using complementary transistors have been designed and fabricated that operate on a one volt power supply and dissipate approximately 6μ watts. The circuits... include a one-shot multivibrator, free-running multivibrator and a set-rest flip-flop."

In another instance it is reported[25] that RCA plans to market a 64 bit scratchpad CMOS memory in late 1970. The access time for this memory is to be 50 nanoseconds and the power dissipation will be 150 nanowatts. Certainly as these kinds of memories become readily available, a digital processor for real time processing will be possible with only a very few constraints.

Judging by the large effort going into speed enhancement and power reduction for Large Scale Integrated circuits it is reasonable to suppose that by 1975, fully addressed memory chips with access times of

100 nanoseconds and power dissipations of no more than 10 μ watts per bit should be available. On this basis, it would be feasible to construct an electronic processor for real time synthetic aperture imaging employing non-quadrature subaperture processing. The space application example which has been referred to throughout this thesis might well prove to be a useful and realistic design. It is summarized below in Table

TABLE VI

Possible Specifications for a 1975 SAR System
with Real Time Digital Processing

<u>Physical Parameters</u>	
Ground swath imaged	40 km.
Satellite altitude	600 n.m.
Satellite velocity	7 km./sec.
Illumination angle of incidence	30 degrees
Minimum differential scattering cross section	-20 db.
Maximum differential scattering cross section	+20 db.
<u>Image Parameters</u>	
Resolution	30 m.
Image M/STD	$\sqrt{18}$
<u>Radar Parameters</u>	
Maximum antenna length (along-track)	4 m.
Antenna width (cross-track)	1 m.
Antenna power gain	40 db.
Receiver Noise Figure	5 db.
Carrier frequency	10.0 GHz
Pulse duration	0.1 μ sec.
Pulse repetition frequency	3500 Hz
Receiver bandwidth	10.0 MHz
Receiver linear dynamic range	40 db.
Peak pulse power	170 kw.
Average transmitter power	60 watts

Subaperture Non-Quadrature Processor Parameters

Number of subapertures to be used (G)	10
Word size in "working stores" (L)	5 bits
Word size in "averaging stores"	9 bits
Total words in "working stores" per range bin	321 words
Total words in averaging stores per range bin	216 words
Total number of range bin (processed in parallel)	1334 bins
Total number of bits in principal stores (averaging + working+buffer)	4.8 megabits
Power to principal stores (10μ watts/bit)	48 watts
Word access time in working stores (time to perform one multiplication)	1.14μ seconds
Word access time in averaging stores	0.43 milliseconds
Power required for digital processor-includes cooling	200 watts
Volume of processor	5 cu. ft.
Weight of processor	60 pounds

SUMMARY AND CONCLUSIONS

This paper was motivated by the question of whether electronic processing of synthetic aperture radar imaging in real time will soon be feasible The most difficult characteristic to achieve in a real-time imaging radar is the truly vast amount of storage which is required within the processor. Thus in a typical space application example, the storage required for matched filtering which is tantamount to SAR processing is 64 megabits for full-focussed quadrature processing. Basically it is the long time during which information is gathered on a ground target (greater than 1 second) that necessitates this amount of storage. Accordingly an investigation of this aspect of the problem revealed that with an image requirement for resolution length greater than the minimum possible a corresponding decrease in the storage requirement was feasible. Thus in the space application example the inherent image resolution was 2 meters, but because the imagery required only 30 meter resolution the storage could be decreased by a factor of 15 to 4.26 megabits.

The processor algorithm to implement digital processing was investigated and it was found that a processor would achieve the same along-track resolution whether or not quadrature processing was employed so that the storage could be cut still further by a factor of 2 to 2.13 megabits. The cost of non-quadrature processing was measured in terms of image quality according to an image figure of merit; the mean-to-standard-deviation ratio (M/STD). The non-quadrature processed image lost about 3 db in M/STD when compared to quadrature processed imagery. The result was an image which appeared to be very grainy. In effect the use of non-quadrature processing maintained resolution in length but lost resolution in gray tone.

The point was made that processing an image of a Rayleigh distributed target for the best resolution possible can give an M/STD ratio no greater than $\sqrt{3.6}$. However, such imagery will be acceptable when the viewer can trade length resolution for increased gray tone tone resolution by "standing back" away from the image. In effect the viewer performs a post detection integration on the grainy image enhancing the effective image M/STD to the viewer by the square root of the area his eye can resolve divided the actual resolution area on the image.

The method by which the storage was cut to produce an image with the desired along-track resolution essentially involved the use of only a portion of the reference functions which are used in full focused processing. Only a limited portion of the doppler spectrum of the return signal was being used to image. Using contiguous portions of the doppler spectrum of the return to create an array of images per range bin which were then summed, the image M/STD was increased by the square root of the number of images summed. The images were uncorrelated with one another and summing them resulted in post-detection integration very much like the way in which the viewer who "stands back" is able to improve his gray tone perception. The method was dubbed "subaperturing" because each uncorrelated image is the result of a synthetic aperture whose main beam is "squinted" in the along track direction so that it does not overlap the other subaperture beams.

"Subaperturing" increased the image M/STD in direct proportion to the square root of the number of subapertures employed, independently of whether quadrature or non-quadrature processing was used. It did not increase the amount of signal storage required in the processor but it did require the inclusion of a large intermediate slow store, the "averaging store" whose size depended on the number of subapertures. Thus in the final analysis the storage required in the processor depended on the image M/STD as well as the resolution length.

Implementation of a non-quadrature subaperture processor employing digital techniques was investigated, and it was found that if each range bin could be processed in parallel as suggested by Khambata, computational time requirements were moderate by today's standards. Implementation via large-scale-integrated circuits of the magnitude of storage required had no precedent as of this writing but the data on LSI-MOS circuits indicated that heat dissipation was a large problem. On the other hand a relatively new branch of LSI technology (CMOS) Complementary Metallic Oxide Semiconductors shows promise of solving the problem of heat dissipation. This technology is still a few years away from being well established.

In conclusion it does appear that an LSI digital processor for Synthetic Aperture Imaging is possible but not practical at this time. The progress of LSI in the CMOS direction is rapid and within the next five years it is very probable that LSI memories with 100 nano-second access times and with power dissipations no greater than 10 μ watts per bit will be readily available. This assumption is the basis for the non-quadrature, subaperture, SAR digital processor design given in Chapter VI.

APPENDIX I

DERIVATION OF RAYLEIGH TARGET DISTRIBUTION

Assume per resolution cell, a large number of scatterers (n) all returning the same voltage amplitude V at random phase angles (ϕ) in the interval $(-\pi, \pi)$. The total amplitude of returned voltage $|V_r|$ is given by:

$$|V_r|^2 = \left[\left(\sum_{i=1}^{n} \cos\phi_i\right)^2 + \left(\sum_{i=1}^{n} \sin\phi_i\right)^2\right] V^2$$

ϕ_i is uniformly distributed on the interval $(-\pi, \pi)$.

The probability density functions for $\cos\phi_i$ and $\sin\phi_i$ are given by:

$$f(\cos\phi_i) = \frac{1}{\pi} \frac{1}{\sqrt{1-\cos^2\phi_i}} \qquad -1 \le \cos\phi_i \le 1$$

$$f(\sin\phi_i) = \frac{1}{\pi} \frac{1}{\sqrt{1-\sin^2\phi_i}} \qquad -1 \le \sin\phi_i \le 1$$

The mean value of $\sin\phi_i$ and $\cos\phi_i$ is zero and the variance of the random variables is one half. Employing the Central Limit theorem and the fact that a squared normal random variable, with a mean of zero and a variance of one, is a chi-square one random variable yields:

$$\frac{2}{n}\left(\sum_{i=1}^{n} \cos\phi_i\right)^2$$

and

$$\frac{2}{n}\left(\sum_{i=1}^{n} \sin\phi_i\right)^2$$

are both chi-square one random variables. Therefore, the return voltage amplitude may be written as

$$|V_r|^2 = V^2 \frac{n}{2}\left[\frac{2}{n}\left(\sum_{i=1}^{n}\cos\phi_i\right)^2 + \frac{2}{n}\left(\sum_{i=1}^{n}\sin\phi_i\right)^2\right]$$

$$= \frac{n}{2} V^2 \left[\chi^2(1) + \chi^2(1)\right]$$

$$= \frac{n}{2} V^2 \chi^2(2)$$

Because the signal backscattered from each point target in the resolution cell is uncorrelated with any other, the average power backscattered from the resolution cell is:

$$P = n\frac{V^2}{2}$$

Therefore,

$$|V_r|^2 = P\chi^2(2)$$

and

$$V_r = \sqrt{P}\sqrt{\chi^2(2)}$$

The mean value of a square root chi-square-two random variable is 1.25 and its variance is 0.43.

APPENDIX II

SYSTEM NOISE MODELING

System noise which arises principally in the front end of the radar receiver is modeled in this program as being Gaussian, narrow band, additive noise. For 4 classes of targets which are described below, the program requires an input indicating which of the four classes of targets is under consideration, a target parameter proportional to the square root of the target scattering cross section, and the desired signal to noise ratio from the receiver front end. If the target belongs to a class of targets other than the four described below, a supplementary running of the program, imaging targets with a given cross section, is required to determine the average power returned to the radar by the targets in the absence of noise. The signal power received at the radar decorrelates when the physical beam is incremented a distance corresponding to one half the inverse of the Doppler bandwidth of the return multiplied by the radar bearing platform velocity. However, the program is written to sample the return power at twice this interval and output all the averaged characteristics of the received signal along with the size of confidence interval for these averages. (If the confidence interval is too large a longer target line length is required.)

The average power of the return from a single class of targets depends on the statistical distribution of the target return from each scatterer of the class within the radar illuminated area. Within the program it is possible to achieve Rayleigh distributed or constant target amplitude return voltages with either constant arbitrary phase or uniformly distributed phase between 0 and 2π. These four classes of targets, which can be internally generated by the program at the option of the user, yield well known power distributions of the received signal and hence require no supplementary running of the program. For example, suppose the average power at the receiver due to the target cross section (σ^2) is given by

$$P = \frac{1}{2} E\{|V_T|^2\} = \frac{\sigma^2}{2} \sum_{i=1}^{m} C_i^2 E\{\chi^2(z)\}$$

$$P = \sigma^2 \sum_{i=1}^{m} C_i^2$$

where E is the expectation operator. The user also specifies the signal to noise ratio of the receiver front end based on Rayleigh distributed targets of parameter σ. The average noise power is given by:

$$P_N = E\{N^2\}$$

where N is a Gaussian random variable of variance σ_N^2 and mean 0.0

$$\therefore \quad \sigma_N^2 = \frac{\sigma^2 \sum_{i=1}^{m} C_i^2}{(S/N)}$$

The program calculates the value of σ_N^2 and alters V_T to be

$$|V_T| = \sigma\left[\left(\sum_{i=1}^{m} C_i \left(\sqrt{\chi^2(z)}\right)_i \cos(\phi_i + \gamma_i)\right)^2 + \left(\sum_{i=1}^{m} C_i \left(\sqrt{\chi^2(z)}\right)_i \sin(\phi_i + \gamma_i)\right)^2\right]^{\frac{1}{2}} + N$$

The value of the random variable N is uncorrelated with its value in succeeding return signals (V_T).

Similarly, σ_N^2 is calculated for each of the 4 possible kinds of targets created by the program according to the following formulas:

2. Targets with constant amplitude return voltage (σ) and uniformly distributed phase

$$\sigma_N^2 = \frac{\sigma^2 \sum_{i=1}^{m} C_i^2}{2\,(S/N)}$$

3. Targets with constant amplitude return voltage (σ) and constant phase (ϕ)

$$\sigma_N^2 = \frac{\sigma^2 \left[\left(\sum_{i=1}^{m} C_i \cos(\phi + \gamma_i) \right)^2 + \left(\sum_{i=1}^{m} C_i \sin(\phi + \gamma_i) \right)^2 \right]}{2\,(S/N)}$$

4. Targets with Rayleigh distributed amplitude voltage returns and constant phase returns

$$\sigma_N^2 = \frac{\sigma^2}{S/N} \left[2 \sum_{i=1}^{m} C_i^2 + (1.25)^2 \sum_{i=1}^{m} \sum_{j=1}^{m} C_i C_j \cos(\gamma_i - \gamma_j) \right]$$

$$i \neq j$$

In the event that external targets are supplied which do not belong to any of these four classes of targets the program must be supplied with a number (A) according to the formula

$$A = E\left\{ \left[\sum_{i=1}^{m} C_i \sigma_i \cos(\phi_i + \gamma_i) \right]^2 + \left[\sum_{i=1}^{m} C_i \sigma_i \sin(\phi_i + \gamma_i) \right]^2 \right\}$$

the imaging system is to be modeled operating over a line of targets with the return voltage from each individual target being Rayleigh distributed in amplitude and uniformly distributed in phase between 0 and 2π. The user specifies that n equally spaced targets in the line of targets lie within the physical antenna's beam. He also specifies the 2-way voltage gain of the illuminating antenna at the point targets (C_i). Since the individual target returns generated by the program are uncorrelated in both phase and amplitude, the receiver voltage amplitude is a random variable given by

$$|V_T| = \sigma\left[\left(\sum_{i=1}^{m} C_i\left(\sqrt{\chi^2(2)}\right)_i \cos(\phi_i+\gamma_i)\right)^2+\left(\sum_{i=1}^{m} C_i\left(\sqrt{\chi^2(2)}\right)_i \sin(\phi_i+\gamma_i)\right)^2\right]^{1/2}$$

where

C_i	is the 2-way voltage gain of the antenna for the i^{th} beam target
ϕ_i	is the phase of the target's reflection coefficient (uniformly distributed between 0 and 2π)
γ_i	is the 2-way round trip phase length between the i^{th} target and the radar system
$\sqrt{\chi^2(2)}$	is the i^{th} target voltage amplitude at the radar (Rayleigh distributed — the square root of a chi squared random variable with 2 degrees of freedom)
σ	is a number specified by user which is proportional to the square root of the differential scattering cross section of the target class.

from which the program will calculate σ_N^2 according to

$$\sigma_N^2 = \frac{A}{2(S/N)} .$$

(A) can be estimated by first imaging a line of homogeneous targets, with the computer program, according to which the signal to noise ratio at the front end of the receiver is to be determined. An estimate of (A) is given as part of the program output as well as a confidence interval for the estimator.

APPENDIX III

SAR COMPUTER SIMULATION PROGRAM DOCUMENTATION

DECK SET-UP TO RUN PROGRAM

```
1         8                       16
$         IDENT                   acct #, user's name
$         OPTION                  FORTRAN
$         LIBRARY                 LB
$         FORTRAN                 NDECK
          {"Fortran subroutine"}
          .
          .
          .
$         FORTRAN                 NDECK
          {Fortran subroutine}
$         OBJECT
          {object deck of subroutine}
$         DKEND
          .
          .
          .
$         OBJECT
          {object deck of subroutine}
$         DKEND
$         EXECUTE
$         LIMITS                  processor time, amount of core,,
                                  lines of output
$         PRMFL                   LB,R,S,7201-BANGERT/R
$         DISC                    01,AIR,10L
$         TAPE                    02,A2DD,,"Tape number",,
                                  "Tape name", OUTPUT
$         TAPE                    08,A3DD,,"Tape number",,
          {data cards}            "Tape name", OUTPUT
$         ENDJOB
```

This deck set-up assumes that the user has rewritten several of the subroutines which are designed for this purpose and is submitting them as FORTRAN source decks while the majority of the program is submitted as object decks. If the user is running only object decks the $ FORTRAN cards may be eliminated; however, the $ OPTION card is still necessary. The set-up also assumes that plotting will be done on the tape with file code 08 using the routines for the Benson-Lehner plotter which are on the BANGERT prmfl. If no plotting is to be done the $LIBRARY, $PRMFL, and $TAPE 08 cards may be eliminated. And last this set-up assumes that at least one data set is to be written on the tape with file code 02 and provides 10 links of disc for the necessary scratch file 01. If no reading or writing of data sets is to occur these two cards may be eliminated.

PRIMARY PROGRAM DATA CARDS FOR RADAR SIMULATION PROGRAM

All ten primary data cards listed below are required inputs and must appear in the order given.

All inputs specified as integer or real must appear right justified in the field given. All real inputs have two assumed decimal places.

Format for Primary Data Cards

1. Target data card - this card fully specifies the number and type of targets to be generated.

Columns	Type of Input	Function of Input
1-6	"TARGET"	identify target data card
18	any character or blank	nonblank indicates data set is to be stored on user supplied tape
19	blank or 1	1 indicates tape is to be initialized with this set, blank means put this set at the end of those already on the tape

Five options are available for generating the targets. One of these five options must be specified starting in column 11. If columns 1-6 of the target card or any one of the other nine primary input cards are blank the program is stopped.

Options:

a) Cycle pattern

Columns	Type of Input	Function of Input
11-15	'CYCLE'	indicates cycle option
21-25	integer	number of targets desired
30-37	'CONSTANT ' or blank	constant, first and third fields following are to be regarded as constants; blank, they are to be multipliers for a Rayleigh voltage distribution
40-45	real	amplitude for this first half of each cycle
47-48	integer	sampling rate for statistical analysis of images. Required input for CYCLE pattern when the number in columns 50-55 and 40-45 are the same
50-55	real	amplitude for last half of each cycle
59-60	integer	number of targets in half of each cycle at low end. Always less than or equal to number in columns 64-65
64-65	integer	number of targets in half of each cycle at high end
69-74	'CPHASE'	if next field contains constant for phase of all targets
	or blank	phase if uniform on $[-\pi,\pi]$
75-80	real	interpreted as constant for phase if 'CPHASE' appears in preceding field, otherwise, ignored. This number must be in range $[-\pi,\pi]$

b) Linear Pattern

11-16	'LINEAR'	specifies linear target pattern
21-25	integer	number of targets to be used
40-45	real	amplitude for first target
50-55	real	amplitude for last target
69-74	'CPHASE'	interprets next field as constant for phase
	or blank	phase is uniform on $[-\pi,\pi]$
75-80	real	interpreted as constant for phase if 'CPHASE' appears in preceding field, otherwise, ignored. This must be in interval $[-\pi, \pi]$

c) Point Pattern

11-15	'POINT'	specifies point pattern
30-37	'CONSTANT'	if next field is to be a constant
	or blank	if next field is to be a multiplier for a Rayleigh voltage distribution with mean of 1.25
40-45	real	constant or multiplier of Rayleigh voltage distribution for amplitude of point target
69-74	'CPHASE'	if next field is to be constant for phase of point target
	or blank	phase is uniform on $[-\pi,\pi]$
75-80	real	interpreted as constant for phase if 'CPHASE' appears in preceding field, otherwise ignored. This must be in the interval $[-\pi,\pi]$

d) User Supplied Targets (user must write subroutine)

11-16	'EXTARG'	specifies user supplied targets through subroutine EXTARG
21-25	integer	exact number of targets to be used (required)
47-48	integer	sampling rate for statistical analysis of images

e) Tape Stored Targets (stored from a previous run or written during current run)

11-14	'TAPE'	specifies tape stored targets
21-25	integer	number of particular data set to be used. This number is assigned at the time the set is written on tape and is output with each use of the data set
47-48	integer	sampling rate for statistical analysis of images

See section on tape storage for restrictions on values to be input on succeeding data cards.

2. Antenna data card - this card fully specifies the characteristics of the antenna to be used.

1-7	'ANTENNA'	identifies antenna data card
11-15	integer	number of resolution cells (i.e., T B product of reference function) in the physical beam of the antenna creating returns
21-25	integer	number of targets per resolution cell
31-35	real	Doppler mismatch referenced to return signal bandwidth

41-45	integer	specifies type of weighting to be placed on antenna elements 1 - square beam, all weights are 1. 2 - cosine on a pedestal (all other entries are in error unless user has expanded subroutine WGTFCN)
51-55	integer	sampling rate for statistical analysis of returns

3. Receiver data card - specifies characteristics of receiver to be used.

1-8	'RECEIVER'	identifies receiver data card
11-15	real	linear gain of receiver over linear range
16-25	real	input saturation point or upper limit (input) of linearity of receiver
31-35	real	logrithmic gain of receiver above saturation point
36-45	real	signal to noise ratio of receiver
46-55	real	constant multiplier of voltage reflection coefficient amplitude on which receiver S/N ratio is to be based
56-65	real	average return amplitude on which to base signal to noise ratio for external targets

4. Processor data card - specifies characteristics of digital processor to be used.

1-9	'PROCESSOR'	identifies processor data card

11-14	'FULL'	full focuss processing
	or 'ZONE'	Zone plate processing
	or 'NONE'	no imaging takes place
20	integer	= 1, cosine channel only used in imaging (non-quadrature processing) = 2, both sine and cosine channels used (quadrature processing)
21-25	integer	number of sub-apertures in physical beam of antenna used in processing returns
31-35	integer	number of resolution cells per sub-aperture
41-45	integer	number of targets incremented per return (PRF)
51-55	integer	number of fully compressed returns overlapped between adjacent sub-apertures
61-65	integer	starting point of first sub-aperture (in fully compressed returns)
71-75	integer	specifies type of weighting to be used on the aperture during processing = 1 - all weights are 1 = 2 - "cosine on a pedestal" all other entries are in error unless user has expanded subroutine WGTFCN

5. APWGTS data card - specifies weights by which each sub-aperture is multiplied prior to summing all for final image line

1-6	'APWGTS'	identifies APWGTS data card
11-20	real	weight for first sub-aperture
21-30	real	weight for second sub-aperture

Addition fields each 10 columns long fill the card. Only as many fields are read as there are sub-apertures. If more than seven sub-apertures are used additional cards with the following format are required. 'APWGTS' appears only on the first card.

Format of Additional Cards

1-10	real	weight for $8n^{th}$ sub-aperture, where this is the n^{th} additional card
11-20	real	weight for 8n + 1 sub-aperture where n as above
etc.		eight of these ten column fields fill each additional card as needed

6. Presum option card - provides for option of summing a given number of returns before they are processed.

1-6	'PRESUM'	identifies presum data card
11-15	integer	number of returns to be summed before imaging occurs, =1 - no presumming <1 or blank, an error

7. Quantization option card - provides for option of quantizing returns and reference function used in imaging.

1-8	'QUANTIZE'	identifies quantize data card
11-15	integer	number of bits available for each word to store the two channels (sine and cosine) of the return
21-25	non-negative real	lower magnitude of range of returns to be quantized
31-35	non-negative real	upper magnitude of range of returns to be quantized
41-45	integer	number of bits available to store sine and cosine reference functions

8. Plot option card - provides for option of outputting original targets and results in such a way that they may be graphed in a plotter.

1-4	'PLOT'	identifies plot data card
11-13	'YES'	plots are to be made according to the next field
	or 'NO'	no plots are to be made
21-24	'ALL'	if previous field is 'YES' results from all sub-apertures and final summations are to be plotted
	or 'NONE'	none of the sub-apertures are plotted, only the final results if 'NO' appears in previous field this field is ignored

If Plot option is taken, user must supply a tape and insert a $TAPE card (see plot write-up).

9. Punch option card - provides option to obtain punched cards for original targets and final results

1-5	'PUNCH'	identifies punch data card
11-13	'YES'	final results are punched
	or 'NO'	final results are not punched
21-23	'YES'	original targets are punched
	or 'NO'	original targets are not punched

No special action is required if this option is taken.

10. Dump option card - provides option to print out all core used as it appears at the end of the processing.

1-4	'DUMP'	identifies dump data card
11-13	'YES'	dump all core storage
	or 'NO'	do not dump all storage

The target amplitude and the results of each sub-aperture and final results are printed in either case.

TAPE STORAGE FOR DATA SETS

This option to store data sets on tape has been provided due to the large amount of computer time necessary to regenerate the targets and returns if they are to be run with several types of processing. Using a tape the targets and returns are generated only once and then stored to be read in when needed. The set may be reread during the run in which they are generated or in a subsequent run.

To place a data set on tape the user must put a non-blank character in column 18 of the 'TARGET' data card. It may be used with any of the pattern options except the TAPE option which reads from the tape. If the tape does not contain any data sets or the sets on the tape are to be destroyed and replaced by the current set. The user must put a '1' in column 19 of the 'TARGET' data card. This initializes the tape. If a '1' does not appear in column 19 and the tape contains other data sets the current set is put at the end of those already there. If no data sets are on the tape a fatal error will halt execution; therefore, '1' must appear in column 19 of the 'TARGET' data card of the first data set to be placed on the tape.

At the time a data set is placed on tape, it is assigned a number. This number is output in the first line of the report for the data set. To access the required data set use the 'TAPE' pattern option and supply the number of the data set right justified in columns 21-25. If the specified data set does not appear on the tape a message to that affect is printed and that data is skipped. Otherwise the set is input into core and processing of the set continues.

The information stored on tape is in two FORTRAN binary logical records. The first logical record contains twenty miscellaneous simple variables. Of these twenty, fourteen are primary inputs to the program and because they are stored on tape they cannot be changed in a subsequent running of this data set. They are fixed. However, any other inputs may be changed. The variables that cannot be changed are: all variables input on the 'TARGET' data card, except the sampling rate for analysis of the images, a number of resolution cells in the antenna beam, the number of targets per resolution cell, the number of targets incremented per return, the number of the antenna weighting function. Values may be inserted for these variables, however, when the tape is read these values will be overwritten. If a data set is to be read from tape error checking on these variables is by-passed thus they may be left blank, error checking for the rest of the data is performed as usual.

The arrays stored on the tape are the target reflection coefficient amplitude, the target phase, the sine and cosine channels signal return amplitudes, the antenna phase, and the antenna weighting function. The sine and cosine channels signal returns are compressed by PRF when they are stored on tape thus it may not be changed, however, the presumming may be changed.

If the user wishes to use the tape storage option he must supply a tape, it may be owned or reserved at the Computation Center. To supply the tape to the program a card similar to the following one must appear after the $EXECUTE card but before the data in the deck set-up.

1	8	16
$	TAPE	02,A1DD,,"tape number",, "tape name", OUTPUT
02	file code, for tape it must be 02	
A1DD		
null		
tape number	number of users tape	
null		
tape name	name of user tape	
OUTPUT	to specify tape may be written on	

Also required for the tape routine is a scratch file, 01, which is usually on disc. The following card is required following the $TAPE card.

	DISC	01,A2R,10L
01	file code for scratch file	
A2R		
10L	10 - specifies number of 3850 word links are required for the file. If a large number of data sets are accumulated on the tape a larger scratch file may be necessary, simply increase the number of links. L-specifies that it is to be a linked file.	

If the user does not wish to use the tape option no action is required. Do not insert the the $TAPE and $DISC cards, however, do not attempt to read or write a data set on tape or the program will abort.

When arranging data sets for a run which involves both reading and writing on the tape the program will accept them in any order, however, the most efficient order is to place all writes first and then the reads later. Also in positioning the reads it is most efficient to put the numbers of the data sets in increasing order. This avoids numerous rewinds or backspaces to reposition the tape.

PLOT OPTION

The plot option has been provided to output the results of this program in a simple and informative manner. Although the numbers for the results are always output it is sometimes difficult to see a pattern in them. The plot option outputs a picture in which it is very easy to compare the results of the processing with the original targets and the results of different types of processing.

The output of the plot option is in the form of an 8.5 by 11 page which is drawn by the Benson-Lehner plotter at the Computation Center from a tape which is created during a run of the program. Contained on the page are five graphs, the top graph is a graph of the target reflection coefficient phase, the second is the target reflection coefficient amplitude, the third is the return signal phase, the fourth is the return signal amplitude and the fifth and last is the image. It may be the image as seen from only one aperture or the final combination of all apertures. The third and fourth graphs are positioned so that as the antenna beam sweeps across the targets (1 and 2) from left to right when the trailing edge of the beam is at target one the return for that antenna position is the first return graphed.

To use the plot option the user should put 'YES' in columns 11-13 of the plot data card, #8. This will result in the plotting of just the final combination of apertures for the final result. If the user also specifies 'ALL' in columns 21-23 a page will be plotted for each sub-aperture. If both options are taken and there are n sub-apertures, then n + 1 pages will be plotted for that data set. The order of the plots will be the first sub-aperture up to the n^{th} followed by the final result.

If the plot option is to be taken the user must supply a tape on which to write the plots. Anything on this tape previously will be destroyed so it would not ordinarily be the same tape as is used for storing the data sets under the tape storage option. The tape must be owned or reserved from the Computation Center. It may not be a scratch tape. The card necessary to provide the tape to program is as follows:

$	TAPE	08,A2DD,,Tape number,, Tape name, OUTPUT

08	plot tape must be on file code 08
A2DD	
null	
tape#	number of users tape
null	
tape name	name of users tape
OUTPUT	specifies that tape is to be written on

It should be remembered that when one plot is created during a run, the one on the tape previously is destroyed so plots should be made as soon after the run as possible to insure that none are accidentally destroyed.

The plot option uses the PLOT-INPLOT plotting subroutines. These routines are available on the system library, however, at present these subroutines do not work. A set of routines by the same name that work are available, however, on a permanent file. To use them the following cards are needed.

```
$      IDENT
$      LIBRARY     LB
       .
       .
       .
$      EXECUTE
$      PRMFL       LB,R,S,7201-BANGERT/R
$      TAPE        08,A2DD,,tape number,,tape name,
                   OUTPUT
       "data"
$      ENDJOB
```

PUNCH OPTION

The punch option was provided so that user might obtain a copy of his targets and results which might be used as input to other programs. One such program which has been used is subroutine PITCHR which prints out pictures of data quantized to thirteen gray levels.

The punch option is exercised as follows: on the PUNCH data place 'YES' in columns 11-13 if the final images are to be punched, place 'YES' in columns 21-23 if the original targets amplitudes are to be punched. These options are independent of each other. The images from each aperture may not be punched.

The format of the punch in both cases is (I10,7F10.3/(8F10.3)). That is, the first card has an integer right justified in columns 1-10, this is the number of numbers to follow, followed by seven real numbers with three decimal places. All other cards contain eight real numbers with three decimal places. If for a particular data set both options are taken the targets are punched before the images.

DYNAMIC STORAGE ALLOCATION

This program has been written using dynamic storage allocation because one of the biggest costs of running a computer job is the charge for the core memory. Using this technique only as much memory as is actually needed must be used.

The method of implementing this technique is as follows: the very first thing done is to call ISPACE, this is a system subroutine which returns as its value the number of memory locations available in blank common. It also arranges that none of this blank common will be used for input-output buffers. In processing a data set the program calculates the lengths of all arrays to be used and plugs these into a formula which also contains the amount available which was returned by ISPACE. The result of this formula is the amount of surplus common available for the current data set. If this amount is negative there is not enough memory available and a message is written stating that the set was aborted due to lack of memory. The process is then repeated, except for the call to ISPACE, for the next data set. When a set is found that will fit in the available memory, processing of the set continues and the next step is to calculate pointers for the arrays which will be used. These pointers indicate the element relative to the beginning of blank common, which is the beginning of the array. There is no unused space left between the arrays. Once these pointers have been calculated, arrays are passed to subroutines as follows:

```
CALL SUB(CELL(IA) , LONG)
```

The subroutine definition would be:

```
SUBROUTINE SUB(CELPHA,LONG)
```

In this example, CELL is the blank common variable and so CELL(IA) is the IAth element of the blank common. In the subroutine SUB that element is taken to be the first element of the array CELPHA. The array CELPHA may then be worked with, without regard to its actual location in the blank common.

In all, fifteen arrays are generated in the blank common region. Of these thirteen are required up to the very end. The other two are needed during only the first half of the program and they are overwritten by arrays generated after they are no longer needed.

To take advantage of this feature the user need only change one card in his program, the $LIMITS control card which follows the $EXECUTE card. The second field of the $LIMITS card specifies the amount of memory to be used for execution avtivity. The ideal situation would be to supply exactly the right number of words so that there is no unused core. Of course, it is impossible to get it that close, but using the following formula the amount of unused core should be below 1024 words, the units in which memory is allocated. This formula should be applied to the largest data set to be run during the current job to determine how much memory will be required.

$$X = \frac{\left(\begin{array}{l} 2(\mathrm{LENREF} + \mathrm{TLONG} + \mathrm{NUMB}) + \text{the maximum of} \\ 2(\mathrm{NUMB} + \mathrm{KANTEL}) \text{ and } 2(\mathrm{NUMA} + \mathrm{L4}) + 3\mathrm{NP} \end{array}\right)}{1024}$$

where

TLONG	number of targets used
KANTEL	number of targets in the physical antenna beam, that is number of resolution cells times targets per resolution cell
NUMB	(TLONG + KANTEL)/INCR
INCR	number of targets incremented per return
NUMA	NUMB/KPRESM

KPRESM	number of returns presummed
LENREF	KANTEL/(INCR*KPRESM)
NP	(KANT*KAN2) (INCR*KPRESM)
KANT	number of resolution cells in a sub-aperture
KAN2	number of targets per resolution cell

In the division above, the value should be increased to the next highest integer if there is any remainder. This value for X is the number of K (1024 word blocks) needed for the blank common. Add to this 19K for the program and this value should be placed in the second field of the $LIMITS card.

MAINLINE

PURPOSE:

To call appropriate subroutines which will perform actual calculations.

ENTRY POINT:

Execution of the program starts with this routine.

REQUIREMENTS:

System routine ISPACE (returns amount of blank common available)
Subroutines SETARG, SETWRP, SNAP, NOISY, PRESUM, QUANT, DIGPRO, and CORDMP.

PROCEDURE:

This mainline first calls system routine ISPACE which returns the amount of blank common available for the current run to the variable MAX. This value will be used to decide if a particular data set can be run. Next it calls a GMAP subroutine SETWRP which sets an address for a wrapup if an abort occurs. This insures that any data sets written on tape are not lost. It next calls SETARG which generates the targets and returns, according to the data cards read in by the module. Upon return from SETARG, it calls SNAP to output the target amplitude and if the dump option is taken it calls SNAP repeatedly to output the antenna phase, the antenna weighting function, and the noise free return amplitude, if noise is to be added later. These are output here as they are destroyed later. NOISY is called next to add any or all of the following to the return signal: system noise, linear gain, and logrithmic gain. If the presum option is taken, PRESUM is called to perform that function. If the quantization option is taken QUANT is called to perform the quantization of the sine and cosine channels of the return. They are treated as one long array. DIGPRO, the digital processor, is called next to process the returns and output the final images unless the no processing option is taken. And last if the dump option is taken CORDMP is called to output all of memory as it appears at the end of processing. The program then loops back to call SETARG which will start the process over again for the next data set.

SUBROUTINE SETARG

PURPOSE:

To coordinate activities to generate targets and returns for radar simulation.

ENTRY POINT:

SETARG

ARGUMENTS:

None

REQUIREMENTS:

Subroutines NGET, OUT2, STAR2, TREADY, ANTENA, BIGPNT, EXTARG, INSPEC, LINEAR, and TWRITE

ERROR FLAGS:

Message is written and data set is skipped if pattern selection is invalid or if data set will not fit in available memory.

PROCEDURE:

The overall function is to generate a set of targets and returns which would be produced at a radar antenna according to information on a set of ten data cards. The first step taken by SETARG is to call subroutine INSPEC which inputs and checks a set of data cards. As soon as an acceptable set is found it returns to SETARG. The TARGET pattern selection is checked and a transfer is made to handle that type of pattern. If the TARGET pattern selection is invalid a message is written and INSPEC is called to input another data set. The processing of each pattern is similar. It involves a call to NGET or NTGET which determines if the data set will fit in available core, if not an error return is taken and the set is aborted. However, if it does fit, NGET calculates the pointers to all arrays required in SETARG. Upon a normal return from NGET the appropriate subroutine is called to generate the actual targets, EXTARG for the external option, STAR2 for the cycle pattern, LINEAR for the linear pattern, and BIGPNT for the point pattern. If the pattern selected was TAPE, subroutine TREADY is called to read the targets from tape and a return is made which skips the next two steps. However, for any pattern except TAPE the next step is a call to ANTENA which generates the returns as seen by a radar antenna. Then the targets, returns (phases and amplitudes) and antenna weighting function are stored on tape, if specified, by a call to TWRITE. SETARG then outputs the first part of the summary which deals with the target selection. OUT2 is then called to print out the rest of the summary dealing with the other input options. This output summary is printed even if the data set will not fit in the available memory so as to identify the data which was aborted. If aborted, control is transferred back to call INSPEC to read in the next data set. If it is not aborted control is returned to the mainline.

SUBROUTINE INSPEC

PURPOSE:

To input and error check a set of ten data cards which will direct the running of the rest of the program.

ENTRY POINT:

INSPEC (TS,IPAT)

ARGUMENTS:

TS	Non-blank if current data set will be stored on tape.
IPAT	number of antenna weighting function.

ERROR FLAGS:

If the card labels in the first six columns are out of order or misspelled a message is written and the input cards are then searched until a 'TARGET' card is found and the input of the next set is begun. The data is error checked for twenty-three errors after it has been input and if any errors are found a copy of the data cards are output along with the appropriate error message or messages.

PROCEDURE:

The first job of INSPEC is to input a series of ten data cards. It does this by inputting a card under a format which is correct for the card. A variable is then set to the number of the card input, that is from one to ten. This variable will be used in a computed go to transfer control to input the next card in the sequence. After each card is input its label is checked to make sure it is the correct card. If it is, control is transferred to input the next card. If at anytime the label does not check out, an error message is written and the program searches for a 'TARGET' data card to start the input of a new set. During this search all cards are read as characters only to avoid bad format error messages. If a card is encountered with columns 1-6 blank, the tape wrap up routine is called and then the program stops. After all ten data cards have been input successfully error checking begins. An array is zeroed out. 'IF' statements test the error conditions and if an error exists the appropriate word of the array is set to one. After all conditions have been checked the array is summed. If this sum is zero no errors exist and control is returned to SETARG. However, if errors were found a copy of each data card is output followed by the error diagnostic. Control is then transferred to the beginning of INSPEC to read in another data set.

SUBROUTINE TREADY

PURPOSE:

To store data sets on tape and read them back.

ENTRY POINTS:

TREADY(IN)

TREAD(K,IPAT,K2)

TWRITE(K,IPAT)

TWRAP

ARGUMENTS:

IN	= 1 tape is to initialized with this set, otherwise, put set at end of those already on the tape
K	Miscellaneous value to be written and read from tape
IPAT	Miscellaneous value to be written and read from tape
*	RETURN 1, normal return for TREAD
K2	Number of data sets to be read from tape
*	Error return if data set requested does not appear on the tape

REQUIREMENTS:

Subroutines NTGET, CALFEF, and FLGEOF

Tape - The user must supply a tape on file code 02 and a scratch file (disc) on file code 01, the scratch file must be long enough to store everything that will be on the tape at the end of the run. Ten links is usually sufficient. If this routine is not to be called these two files are not required.

ERROR FLAGS:

If an attempt is made to initialize the tape twice during one run the second one is ignored with a message written. If while copying from the tape to the scratch file a data set is encountered which cannot be read into the available memory an error message is written to the effect that writing is impossible on the tape unless memory is increased by so much, however, the tape may still be read as usual.

PROCEDURE:

TREADY has four distinct parts each of which handles a different phase of the tape I/O. In the following, the word tape will be used to mean either tape or disc whicheaver one happens to be the write file.

The first phase is handled by entry at TREADY. This positions the tape so that it may be written on. If the argument IN is one the tape is initialized with this set. This may be done only once during a run. If IN is not one, the tape is positioned at the end of the other sets on the tape, this may involve copying from one file to another. When the file has been positioned so that it may be written on an indicator is turned on.

The second phase is handled by entry at TWRITE. TWRITE writes two FORTRAN logical records on the tape in the binary mode. The first record contains 20 words which are simple variables associated with the data set, including a sequence number for its position on the tape aa the first word. This is the number used to find the set when it is to be read back later.

TREAD handles the third phase which is reading a particular data set from the tape. All that is necessary is the number of the data set. The tape is searched, using the first word of the first logical record which is the data set number, until the proper set is found. If it is not found a message is written and an error return is taken. Normally, however, the first record is then read and NTGET is called to determine if the data set will fit, if not the second record is not read and an error return is taken. Otherwise, the second record is read and CALREF is called to calculate the reference function to be used in imaging. It then returns.

The fourth phase is the wrap up. Because the data must be copied back and forth between the tape and the scratch file at the end of the program the good copy might be on the scratch file. TWRAP determines where the good information is and if need be copies it back to the tape.

COMMENTS:

In order that the time used in writing on the tape be cut to a minimum it would be advantageous to copy the data only twice, once to the scratch file and then back to the tape, if writing is to occur. This may be done by placing all data sets that will write on the tape before all sets that will read from the tape in the data deck. Also when reading it is most efficient to place data set numbers in increasing order. However, the program will accept them in any order.

PURPOSE:

To determine if current data set will fit in available memory and calculate points used in dynamic allocation.

ENTRY POINT:

NGET(NUMB,TLONG,KANTEL,*)

NTGET(NUMB,TLONG,KANTEL,*)

ARGUMENTS:

NUMB	number of returns to be generated
TLONG	number of targets used
KANTEL	number of targets in physical beam of antenna
*	error return if data set will not fit in available memory

REQUIREMENTS:

None

RESTRICTIONS:

None

PROCEDURE:

First the lengths of several arrays are calculated. Then using these lengths and the amount of blank common available, the amount of surplus is calculated. If this number is negative, an error return is then taken. Otherwise, the pointers for the arrays to be used in subroutine SETARG are calculated. Finally a variable is set to 1 indicating statistical analysis is to be performed when the cycle pattern is specified and the amplitude over both halves of the cycle is the same (constant or statistical).

SUBROUTINE STAR2

PURPOSE:

To generate targets for the cycle and point target pattern options.

ENTRY POINTS:

STAR2(CELPHA,CELAMP,I27)

BIGPNT(CELPHA,CELAMP,KAN,I27)

ARGUMENTS:

CELPHA	Array for phase of targets
CELAMP	Array for amplitude of targets
I27	Number of targets, length of arrays
KAN	Number of targets in physical antenna beam

COMMON STORAGE (only variables used are listed):

1) PXCH/

CPC1	Number of targets in one half of cycle at low end
CPC2	Number of targets in one half of cycle at high end
HI	Amplitude or multiplier of square root chi-square two random variable for targets in first half of cycle
LOW	Amplitude of multiplier of square root chi-square two random variable for targets in last half of cycle
S2	= 'CONSTA' target amplitudes of reflection coefficients are constant otherwise Rayleigh distributed
CHANGL2	= 'CPHASE' phase target reflection coefficient is constant otherwise uniform on $(-\pi,\pi)$
ANGL2	Constant for phase
NTIMES	Number of cycles of each size

ERROR FLAGS:

None

REQUIREMENTS:

System routines RCM (uniform random number generator), RMS (normal random number generator), SQRT (square root)

PROCEDURE:

Subroutine STAR2 has two distinct functions: 1) generating target reflection coefficient amplitudes and phases for the CYCLE pattern option by entering at STAR2; and 2) generating a target reflection coefficient amplitude and phase for the point pattern option by entering at BIGPNT.

STAR2 consists of three nested DO loops. Since STAR2 generates a cycle pattern the outer loop controls the size of the cycle being generated. The next loop controls the number of cycles of each size that will be generated. The inner-most loop where the calculations are performed is iterated from one to half the size of the current cycle. This loop is traversed twice for each cycle changing the multiplier for the amplitude in the meantime. The count for the actual target to be calculated is kept by a single variable which is incremented in the inner most loop.

BIGPNT handles the point target pattern. The number of targets used depends entirely on the size of the antenna being used and is equal to two times the size of the antenna (in targets illuminated by the physical antenna beam in the along-track direction) plus one. If N targets are in the physical antenna's beam, the first and last N targets in the target line have zero reflection coefficient amplitudes. The center target has a reflection coefficient amplitude other than zero.

BIGPNT first zeros out both the phase and amplitude of the targets, it then calculates either a constant or statistical phase and amplitude of reflection coefficient which is stored as the very center target.

SUBROUTINE LINEAR

PURPOSE:

To generate targets for the linear target pattern option

ENTRY POINT:

LINEAR(CELPHA,CELAMP,M)

ARGUMENTS:

CELPHA	Array for target phase
CELAMP	Array for target amplitude
M	Number of targets, length of arrays

COMMON STORAGE(only variables used are listed):

1)	PXCH	
	FIRST	Amplitude of first target reflection coefficient
	LAST	Amplitude of last target reflection coefficient
	CAN	= 'CPHASE', phase will be constant otherwise, phase is uniform over $(\pi,-\pi)$
	A47B	if CAN = 'CPHASE', constant for phase otherwise ignored

ERROR FLAGS:

None

REQUIREMENTS:

Routine RCM (random number generator, uniform over (0.1))

RESTRICTIONS:

None

PROCEDURE:

The slope of the amplitude line is calculated using the number of targets and the amplitude for the first and last target which are inputs. The first target is assigned its amplitude. The slope is added to it to get the value of the second target and so on until the final target is reached. The phase is assigned as a constant value if CAN = 'CPHASE' otherwise a uniform random number is generated for each target using RCM and the phase is uniform over $(-\pi,\pi)$.

SUBROUTINE EXTARG

PURPOSE:

To allow the user flexibility in inputting his targets

ENTRY POINT:

EXTARG(CELPHA, CELAMP, L, K)

ARGUMENTS:

CELPHA	Array for phase of targets
CELAMP	Array for amplitude of targets
L	Number of targets, length of arrays
K	= 5 this is dummy program
	= 4 this is user supplied program

COMMON STORAGE (only variables used are listed):

1) PRO2/

KSAMPD	Sampling rate for statistical analysis of images
KSAMPT	Sampling rate for analysis of targets
KSAMPR	Sampling rate for analysis of returns
LOOPY	= 1 statistical analysis of targets, returns, and images will be performed
	≠ 1 no statistical analysis performed

ERROR FLAGS:

If this subroutine is called an error message is written and the current data set aborted. User must write his own subroutine.

REQUIREMENTS:

None

RESTRICTIONS:

The value of L must not be changed in EXTARG. K must have the value 4 upon exit from the user written EXTARG.

PROCEDURE:

This subroutine in its current form writes an error message and returns with K = 5 which will abort the current data set.

COMMENTS:

The subroutine in its current form should never be called. The purpose of this routine is to serve as a place holder for a subroutine which a future user may write to input his targets. This will give the user flexibility to input types of targets not covered as standard options.

There are many options open to the user in writing this routine to generate his targets. He may use different combinations of the system random number generators than those provided as standard options, or he may input them via cards or tape. When they are to be input by cards on file 05 along with the program data cards, the data cards to be read in by EXTARG should follow immediately after the last program data card for that particular set. A word of warning, EXTARG is not called if an error is found in the program data cards or if the data set will not fit in the available memory.

It may sometimes be necessary to perform statistical analysis on the targets, returns, and images for the external option. The user must input sampling rates for the analysis. The sampling rate for the images is input in columns 47-48 of the 'TARGET' data card and for the returns in columns 54-55 of the 'ANTENNA' data card. The sampling rate for the targets is assumed to be one unless increased in EXTARG. It is the variable KSAMPT, the sixth position in labeled common PRO2.

SUBROUTINE ANTENA

PURPOSE:

To calculate returns so as to simulate a radar antenna.

ENTRY POINTS:

ANTENA(CELPHA,CELAMP,REAMPS,REAMPC,ANTPHA,REAMP,
REPHA,WGT,J7,J57,TLONG,IPAT,SINANG,COSANG)

CALREF(ANTPHA,SINANG,COSANG,J7)

ARGUMENTS:

CELPHA	Target reflection coefficient phase
CELAMP	Target reflection coefficient amplitude
REAMPS	Sine channel of return
REAMPC	Cosine channel of return
ANTPHA	Antenna phase
REAMP	Return amplitude
REPHA	Return phase
WGT	Antenna weighting function
J7	Number of targets in full physical antenna beam
J57	Number of returns to be calculated
TLONG	Number of targets used in target line
IPAT	Number of antenna weighting function to be generated
SINANG	Sine reference function
COSANG	Cosine reference function

PROCEDURE:

ANTENA first calculates an antenna phase for the particular antenna being used. The antenna phase for any particular target in the physical beam is a function of the position of the beam, the number of resolution cells in the beam and the number of targets per resolution cell. The sine and cosine reference functions are then calculated based on the antenna phase and the doppler mismatch, if any. Generating these two arrays is the sole purpose of the entry point CALREF. ANTENA next calls subroutine WGTFCN which generates the antenna weighting function according to the number input on the 'ANTENNA' data card for this purpose. A weight is generated for each target in the antenna beam. The loop that makes up the rest of ANTENA generates four arrays, the sine and cosine channel of the return and from both of these the whole return amplitude and return phase. The sine channel of the return is the sum of the reflection coefficient amplitude of the target times the weighting function for the position of the target seen by the antenna in its current position. The antenna is moved according to an increment value input by the user. Thus, the antenna may step one target between returns or ten targets; whatever is desired. Only the returns necessary for that increment value are calculated as this is a very time consuming part of the program. The whole return amplitude is the square root of the sum of the squares of the two channels. The return phase is the arctan of the sine channel divided by the cosine channel. If the cosine happens to be zero it becomes the sign of the sine channel times $\pi/2$.

PURPOSE:

Calculate specified weighting function for antenna and processor. Left open for user expansion.

ENTRY POINT:

WGTFCN(WGT,N2,IPAT)

ARGUMENTS:

WGT	Array for weighting function
N2	Length of weighting function
IPAT	Specifies type of weight to be generated = 1, weight of 1, for all elements = 2, "cosine on a pedestal"

ERROR FLAGS:

If number of weighting functions specified is greater than options available weighting function number one is assumed with error message written.

COMMENTS:

At present this subroutine has only two options for the weighting function: 1) a weight of 1 for each element of the array; and 2) cosine on a pedestal. However, using the following instructions the user may expand this subroutine to include his own weighting functions.

1) Change value of MAXPAT in data statement to agree with the number of functions now available.
2) Change the computed go to statement labeled 100 to have as many arguments as there are functions. The arguments added will be the statement numbers of where each function begins.
3) Write the function desired, the first statement of which must contain a label which appears in the computed go to of statement 100 and the last statement must be a RETURN.

To access the new weighting function input as the weighting number the position of the label of the function in the computed go to statement 100.

If the user wishes to input weights via cards on the system input file, 05, these cards should come after the last required program data card and after any cards that will be input by subroutine EXTARG, if user inputs targets by cards. If cards are to be read for both the antenna and reference function weighting the order should be: EXTARG targets, antenna weights, reference function weights.

In writing a weighting function for the antenna a value must be supplied for every target position in the physical beam. This is equal to the number of resolution cells times the number of targets per resolution cell. The length of the weighting function that must be provided for the reference function is the number of targets in the sub-aperture being used. This is equal to the number of resolution cells in the sub-aperture times the number of targets per resolution cell. This is mentioned because if the functions are to be input to the program, rather than calculated internally, these values must be known prior to running the program.

SUBROUTINE CORDMP

PURPOSE:

To output a summary of the data set and all of core.

ENTRY POINT:

CORDMP

SNAP(ARRAY,HEAD,LONG)

OUT2(IPAT)

ARGUMENTS:

ARRAY	Locations of one array which is to be output
HEAD	A three word name which will appear at beginning of ARRAY on output sheet
LONG	Length of ARRAY
IPAT	Number of weighting function for antenna

PROCEDURE:

Each entry point has an entirely different function, however, each deals with output. Entry at CORDMP outputs all of working memory by arrays. Each array is headed by a name identifying it. Entry at SNAP will output the one array and heading which are agruments.

OUT2 outputs a summary of the current data set except for the part dealing with the generation of targets. All variables which are output are in common, except IPAT.

SUBROUTINE NOISY

PURPOSE:

To add noise to the returns, if specified, and to add linear and logrithmic gain if specified, that is to simulate a radar receiver.

ENTRY POINT:

NOISY(REAMP,REPHA,REAMPS,REAMPC,ANTPHA,WGT1,NUMB, KANTEL,CELPHA)

ARGUMENTS:

REAMP	Array containing return amplitude
REPHA	Array containing return phase
REAMPS	Array containing sine channel of return
REAMPC	Array containing cosine channel of return
ANTPHA	Array containing phase of antenna
WGT1	Array containing weighting function of antenna
KANTEL	Number of targets in physical beam of antenna
CELPHA	Array containing target phase

COMMON STORAGE (only variables used are listed):

1) RECEIV/

GAIN	Linear gain of receiver
SAT	Upper magnitude (of input signal) of linear range, clipping point
ALGAIN	Logrithmic gain of receiver for input value above clipping point
STN	Signal to noise ratio for receiver

2) OPTION/

SELECT	Pattern for amplitude of original targets
RCA	Mean voltage reflection coefficient amplitude
AVEAMP	if SELECT = 'EXTARG', average value of return amplitude (REAMP)

3) PRO2/

LENUNC Number of returns in physical antenna beam along-track length

KSAMP2 Sampling rate for statistical analysis of return amplitude

LOOPY = 1 if statistical analysis is to be performed

$\neq$ 1 no statistical analysis

ERROR FLAGS:

None

REQUIREMENTS:

System routines - SIN, COS, RMS (normal random number generator, mean O, standard deviation 1), SQRT (square root), ATAN2 (arc tangent), ALOG10 (common logrithm), SUBROUTINE - MENVAR (for statistical analysis)

RESTRICTIONS:

If EXTARG pattern option is taken and noise is to be added, user must supply the average value of return amplitude as the value of AVEAMP if this is omitted no noise will be added (see EXTARG inputs).

COMMENTS:

If certain options are not to be taken, variables should be input as follows: if a straight linear receiver is desired set GAIN = 1., SAT = 999. and ALGAIN = 0. If no noise is to be added set STN = 999.99 (these values are input on RECEIVER data cards).

PROCEDURE:

If noise is to be added to the returns a multiplier is first calculated which depends on the distribution of the amplitude and phase of the targets, the input signal to noise ration, and the input mean voltage reflection coefficient amplitude. There are five cases each with a different formula for calculating this multiplier. They are as follows:

(S/N) = input signal to noise ratio (STN)

σ = mean voltage reflection coefficient (RCA) amplitude

$C_i = i^{th}$ element of antenna weighting function (WGT1)

RNOISE = multiplier for adding noise (RNOISE)

n = number of targets in antenna

γ_i = i^{th} element of the antenna phase

ϕ = constant target phase

Case 1: Statistical amplitude and statistical phase

$$\text{RNOISE} = \sqrt{\frac{\sigma^2 \sum_{i=1}^{n} C_i^2}{S/N}}$$

Case 2: Constant amplitude and statistical phase

$$\text{RNOISE} = \sqrt{\frac{\sigma^2 \sum_{i=1}^{n} C_i^2}{2\ S/N}}$$

Case 3: Statistical amplitude and constant phase.

$$\text{RNOISE} = \sqrt{\frac{\sigma^2}{2\ S/N} \left[2\sum_{i=1}^{n} C_i^2 + (1.25)^2 \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} C_i C_j \cos(\gamma_i - \gamma_j) \right]}$$

Case 4: Constant amplitude and constant phase

$$\text{RNOISE} = \sqrt{\frac{\sigma^2 \left(\left(\sum_{i=1}^{n} C_i \cos(\phi + \gamma_i) \right)^2 + \left(\sum_{i=1}^{n} C_i \sin(\phi + \gamma_i) \right)^2 \right)}{2\ S/N}}$$

Case 5: External targets

$$\text{RNOISE} = \sqrt{\frac{A}{2\ S/N}}$$

where A = average return amplitude input by user

Once this multiplier has been computed system noise is added to the single channel of the return according to the following formula:

$$\text{REPHA}_a = REPHA_b + \arctan \frac{A_1}{REAMP_b + A_2}$$

$$\text{REAMP}_a = \sqrt{(REAMP_b + A_2)^2 + A_1}$$

where A_1 and A_2 are distinct normal random numbers.

If the signal to noise ratio has been specified as 999.99 the preceding skipped. Next the signal is amplified by the linear and logrithmic gain and then the single channel is split into the sine and cosine halves. If statistical analysis is indicated on returns it is done at this point using the sampling rate input in columns 54-55 of the 'ANTENNA' data card. The analysis is performed on the single channel of the return and it ignores both the first and last antenna length of returns.

SUBROUTINE PRESUM

PURPOSE:

To sum a desired number of elements of two sequential arrays and compress them as they are being summed.

ENTRY POINT:

PRESUM(REAMPS,I)

ARGUMENTS:

REAMPS	Location of first of two sequential arrays to be pre-summed.
I	Number of sequential to be summed

COMMON STORAGE (only variables used are listed):

CELENG/

NUMB	Length of arrays before presum
IF	Pointer to first array presummed
IG	Pointer to second array presummed
IH	Pointer to array following second array presummed

ERROR FLAGS:

None

REQUIREMENTS:

None

RESTRICTIONS:

None

PROCEDURE:

The first I elements of the first array are summed and the value stored in element one of that array. Then the next I elements are summed and the value stored in the second element of the array and so on for the first array, its size being reduced by a factor of I. If on the last round for that array I elements are not available to be summed the ones that are left are discarded. Then the first I elements of the second array are summed and its value is stored in the location immediately following the last location used by the now compressed first array and so on until the second array is completed. The pointers to the second pre-summed array and the one following it are recalculated using the lengths of the now compressed arrays.

SUBROUTINE QUANT

PURPOSE:

To quantize an array to integer values (of real type) according to the number of bits available for a computer word to store the value.

ENTRY POINT:

QUANT(REAMPS,NTWICE)

ARGUMENTS:

REAMPS	Array to be quantized
NTWICE	Length of array REAMPS

COMMON STORAGE:

1) OPTION

IQUANT	Number of bits available to store value
RANG1	Lower magnitude for range to be quantized
RANG2	Upper magnitude for range to be quantized

ERROR FLAGS:

None

REQUIREMENTS:

None

RESTRICTIONS:

$0 < \text{IQUANT} < 36$

$0 < \text{RANG1} < \text{RANG2}$

PROCEDURE:

The quantization is performed on the absolute value of the elements of the array over the interval specified by the limits of quantization. The original sign is then put back on. If the absolute value falls outside the interval it is truncated back to the nearest endpoint. Otherwise, it is given the value of the largest integer in the difference of the element and the lower limit of quantization divided by the size of each quantum. The size of the quantum equals the difference of the range limits divided by one less than the number of positive quantization levels. The number of quantization levels equals 2.**(IQUANT), when IQUANT is the number of bits available to store the quantized value. Of these levels half are positive and half are negative with both a positive and a negative zero. Thus the maximum positive value after quantization is 2.**(IQUANT-1).

SUBROUTINE DIGPRO

PURPOSE:

To simulate a radar digital processor and image a set of targets from a series of returns.

ENTRY POINT:

DIGPRO(SINANG,COSANG,CELPHA,CELAMP,REAMP,REPHA,
REAMPS,REAMPC,AMAP,AMAPS,SPROD,CPROD,WGT2,L1,L3,
L4,KTGS,L2,NUMB)

ARGUMENTS:

SINANG	Sine reference function
COSANG	Cosine reference function
CELPHA	Target phase
CELAMP	Target amplitude
REAMP	Return amplitude
REPHA	Return phase
REAMPS	Sine channel of return
REAMPC	Cosine channel of return
AMAP	Image for one aperture
AMAPS	Final image
SPROD	Intermediate value used in imaging
CPROD	Intermediate value used in imaging
WGT2	Weighting function for each sub-aperture
L1	Total number of returns
L3	Number of returns per sub-aperture
L4	Number of images to be generated
KTGS	Total number of original targets
L2	Number of returns in one antenna length
NUMB	Total number of returns before any presum

REQUIREMENTS:

System routine SQRT (square root) Subroutines MENVAR,
OURPLT,WGTFCN

PROCEDURE:

DIGPRO's function is to simulate a digital processor. It is designed to handle all of the following: sub-apertures which may or may not overlap, full or zone focused processing (the zone-focused uses only the sign of the reference function), and quantizing of the weighted reference function. DIGPRO first calls WGTFCN which generates a weight for each target in the sub-aperture being used. All of these weights are not necessary, however, because the returns which they will be lined up with may have been compressed by pre-summing or by skipping more than one target when generating the returns initially, therfore, they are compressed by the product of the two above numbers. If zone focusing is used the elements of the sine and cosine reference functions are transformed into a + 1 or - 1 depending on the sign of the element. For the imaging, a DO loop controls the aperture being worked on. Two pointers are first calculated, one for the returns and one for reference function, which depends on the particular aperture being imaged. Two intermediate arrays, the weighted reference functions, are calculated and then quantized if this is indicated. Each pass through the next DO loop images one target and adds this image with the APWGTS weighting factor for this aperture to the images from other apertures to get what will be the final results. As soon as every target has been imaged using the current aperture, these images are output, statistically analyzed if indicated and then plotted if the plot option for the sub-apertures has been taken. The next aperture is then imaged with the only difference being that the pointers to the returns and the reference function have different values which results in an image as seen from the new aperture. After all apertures have been imaged, the final result is printed, analyzed if indicated, and plotted if the plot option was taken. The final image is also punched if the punch option is taken. It then returns to the mainline.

SUBROUTINE OURPLT

PURPOSE:

To graph five arrays, one above the other in the form of an 8.5 by 11. inch page.

ENTRY POINT:

OURPLT (CELPHA, CELAMP, REPHA, REAMP, AMAP, KTAREL, L4, NUMB, LENUNV, J)

ARGUMENTS:

CELPHA	First (top) to be graphed
CELAMP	Second array to be graphed
REPHA	Third array to be graphed
REAMP	Fourth array to be graphed
AMAP	Fifth array to be graphed
KTAREL	Length of CELPHA and CELAMP
L4	Length of AMAP
NUMB	Length of REPHA and REAMP
LENUNC	First element of REPHA and REAMP to be plotted
J	= 1 for first entry with current arrays CELPHA, CELAMP, REPHA, and REAMP >1 for not first entry with current first four arguments

ERROR FLAGS:

None

REQUIREMENTS:

System routines PLOT and INPLOT

Subroutine LNEPLT

Tape User must supply a tape with file code 08 on which to write the plots, if this routine is to be called.

RESTRICTIONS:

Values for the arrays CELPHA must have magnitude no greater than 3.2 (It is only when user generates his own targets in subroutine EXTARG that this must be watched).

PROCEDURE:

First the maximum of the arrays are found. CELPHA and REPHA are known to be in the interval -3.2 to 3.2. The other arrays have a minimum of zero. Since this routine may be called more than once with the same first four arguments the maximum for them are only calculated the first time. These limits are used in scaling the plots. An 8.5 x 11. inch border is drawn and then LNEPLT is called to plot each array. After the last call to LNEPLT the pen is repositioned for the next page of plots.

SUBROUTINE LNEPLT

PURPOSE:

To graph one array

ENTRY POINT:

LNEPLT (XMIN, XMAX, YMIN, YMAX, ARRAY, I1, I2, DOWN)

ARGUMENTS:

XMIN	Minimum X coordinate to be plotted
XMAX	Maximum X coordinate to be plotted
YMIN	Minimum Y coordinate to be plotted
YMAX	Maximum Y coordinate to be plotted
ARRAY	Array to be plotted
I1	First element of array to be plotted
I2	Last element of array to be plotted
DOWN	Downward length for pen to be moved after completing the plot (repositions for next plot)

ERROR FLAGS:

None

REQUIREMENTS:

Routines System plotting routines PLOT and INPLOT

RESTRICTIONS:

None

PROCEDURE:

INPLOT is called to set the limits for the plot. All plots are 6 inches long and 1.4 inches high. The array is plotted by repeated calls to PLOT. After the array has been plotted the X axis is drawn and the pen positioned downward for the next plot.

SUBROUTINE MENVAR

PURPOSE:

To perform statistical analysis on the array, that is to find the mean, variance, and covariance of the array.

ENTRY POINT:

MENVAR (KSAMPD, K1SAMP, ARRAY, HEAD, KFSAMP)

ARGUMENTS:

KSAMPD	Sampling rate to be used in analysis
K1SAMP	First element of array to be used
ARRAY	Array to be analyzed
KFSAMP	Last element of array to be used
HEAD	Two Word Label to be output

REQUIREMENTS:

SQRT (square root)

PROCEDURE:

Using the numbers of the first and last sample to be used, the sample is divided into two equal parts. The average of each part and the average of the elements squared in each part are calculated. These four numbers are then used in place of the whole array in finding the mean, variance, and the covariance of adjacent samples.

The mean and variance are calculated in the usual way and a T-test determines a 90% confidence interval. The mean squared divided by the variance is also calculated.

SUBROUTINE SETWRP

PURPOSE:

To set a wrap up address to be used in case of an abort.

ENTRY POINT:

<u>SETWRP</u>

PROCEDURE:

SETWRP is a GMAP subroutine. It first gets the FORTRAN wrap up address from the upper half of word 23 (decimal) of the program and saves it in symbolic location GOTO. It then gets the address of symbolic location WRAPUP and stores it in the upper half of word 23. It then returns to the calling program. In case of an abort GECOS transfers control back to the user at the address contained in word 23 which is now WRAPUP. The instruction at this location is a call to routine TWRAP which is the last routine called if termination is normal. When TWRAP is finished it returns and the next statement is at location GOTO which transfers to the original wrap up address.

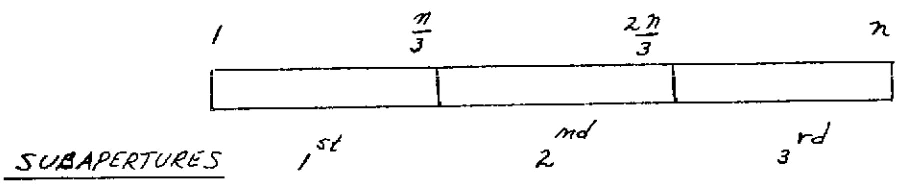
DUMP OPTION

The purpose of the DUMP option is to allow the user to output all arrays which are used during the processing of one data set. The dump option is taken by placing 'YES' in columns 11-13 of the DUMP data card. The arrays output in order are the following: antenna phase, antenna weighting function, sine reference function, cosine reference function, target phase, target amplitude, return amplitude, return phase, sine channel of return, cosine channel of return, processor weighting function, weighted sine reference function for last aperture imaged, weighted cosine reference function for last aperture imaged, final image, image for last aperture. The first two arrays are output separated from the rest because they no longer exist when the rest are output. Each array is headed by a title and the length of the array. The array is output ten numbers to a line. If the dump option is not taken the target reflection coefficient amplitude, the image of each aperture, and the final image is still output. If the dump option is to be used on large data sets it should be remembered that a large amount of output will be generated and it may be necessary increase the output line limit. The standard limit for the execution activity is 5000 lines. This limit may be changed by placing the desired number in the fourth field of the $LIMITS cards following the $EXECUTE card. 10,000 is the maximum number of lines that may be output by a normal job.

APWGTS DATA CARD

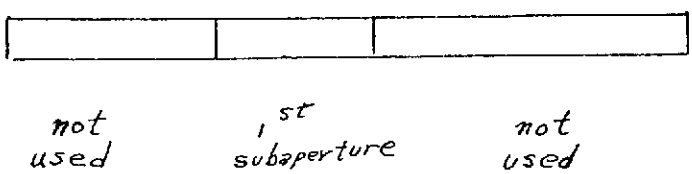
'APWGTS' stands for aperture weights. A weight must be supplied for each aperture used in the imaging. The number of apertures was input of the previous card. The weights may be zero or negative, as well as positive, however, they may not all be zero. The aperture weights are input seven to the first card and eight on all succeeding cards until the correct number has been input. Only the first card has 'APWGTS' in columns 1-6. All numbers are in fields ten columns wide.

In forming the final result a weighted sum is performed on the results of each aperture with these as the weights. Thus, an aperture may be eliminated from the final result by making its weight zero. In supplying weights for the sub-apertures they should be viewed as follows:



where n is the total number of nominal resolution cells in the physical antenna beam

The above diagram shows an antenna beam of n resolution cells with three sub-apertures. The first sub-aperture uses only the first third of the reference function, number two uses the middle third, and number three the last third. However, it might be that only one sub-aperture is used as follows:



In this case only one weight need by provided because only one aperture is used, even though it is in the middle of the physical beam.

PRESUM OPTION

The function of the presum option is to sum series of returns of specified length prior to the imaging of those returns. This cuts down the number of memory locations needed during imaging. To take this option place the number of returns in a series to be summed in columns 11-15 of the 'PRESUM' data card. If no presumming is to take place a one must appear in that field. This presumming occurs after the signal has been through the receiver and picked up noise and been amplified, but before the return is quantized. If the number to be presummed does not divide evenly into the total number of returns the few returns remaining at the end of the array are discarded.

QUANTIZATION OPTION

This option gives the user the ability to specify the size of the word that will be used to store the return and the weighted reference function used in the processor. The size is given in a certain number of bits. If n bits are specified the number of quantization levels will then be 2^n. This is really two options as a separate word size is supplied for the returns and the reference function. The size of the word may be from 1 bit to 36 bits (number of bits per wood on the GE-635). To take the return quantization option place the number of bits in columns 11-15 of the 'QUANTIZE' data card (a number must always appear in this field) and specify the range of the quantization by placing the lower limit in columns 21-25 and the upper limit in columns 31-35. This range is for the return signal prior to quantization but after it has been amplified by the receiver and presumming has taken place. The lower limit must always be less than or equal to the upper limit and it must be non-negative. If 36 bits is specified the range specifications may be left blank. The number of bits for the reference function is specified in columns 41-45 (a number must always appear in this field). No range is necessary as it is always between - 1 and + 1. Note that the quantization range specified is over absolute values. If the range were specified from 20 to 100 in 5 bits then the ranges - 100 to - 20 and 20 to 100 would each be quantized into 16 equal levels.

STATISTICAL ANALYSIS OPTION

This option allows the user to obtain some statistical data on his targets, returns, and images. The output of each analysis contains a heading which tells what is being analyzed, targets, returns, image of current aperture, or final image. It also contains the sampling rate employed in the analysis and the number of samples used. The results of the analysis are then output. They include the following: the mean image voltage of the sample, the mean image voltage variance, the mean covariance, and the mean squared voltage divided by the voltage variance. Each of the above is accompanied by a 90% confidence interval which is obtained by a T-test. Statistical analysis takes place whenever one of the following occurs: 1) if the CYCLE pattern option has been specified (or read from tape) and the user supplied constant is the same for both halves of the cycle; or 2) the user has supplied a sampling rate for the returns (columns 51-55 of the 'ANTENNA' data card) or for the images (columns 47-48 of the 'TARGET' data card). These sampling rates are positive integers. Really (1) above is a special case of (2) because if the conditons of (1) are met a sampling rate must be provided for the analysis of the images or it is a fatal error. The targets are always analyzed with a sampling rate of one because all internally generated targets are uncorrelated. If the user inputs correlated targets through subroutine EXTARG this sampling rate may be changed. The directions for this are in subroutine EXTARG. It should be noted that since this procedure requires a change in a subroutine of the program, if the targets are stored on tape to be read in later it will be impossible to change the sampling rate from one when they are read in. Thus they must be analyzed when they are created. The sampling rates for the returns and images may be changed at any time, however, one other important fact, analysis of returns takes place after noise has been added to the returns and/or the return has been amplified.

CYCLE TARGET PATTERN OPTION

The cycle pattern is one of the three internally generated types of targets. It consists of a series of highs and lows (of target reflection coefficient amplitudes) of equal size. There are four possible combinations of target amplitude and phase for this option. These are: 1) statistical amplitude and phase; 2) statistical amplitude and constant phase; 3) constant amplitude and statistical phase; and 4) constant amplitude and constant phase. In the cases with constant amplitude, there is one constant for the first half of each cycle and a second constant for the second half of each cycle. These numbers may be the same or different. Statistical amplitudes have a Rayleigh distribution with a mean of 1.25 times the user supplied constant. The statistical phases are uniformly distributed on $(\pi, -\pi)$.

To take the CYCLE option the user must place the word 'CYCLE' in columns 11-15 of the TARGET data card. He must then specify an approximage number of targets to be used in columns 21-25. This number is only approximate as will be explained later. He must then supply a number for each half of the cycle. The first half in columns 40-45 and the second half in 50-55. If the targets are to be constant rather than statistical the word 'CONSTANT' must appear in columns 30-37. If the number supplied for both halves is the same, a number must appear in columns 47-48. This is the sampling rate for the statistical analysis of the images which will occur. The user must also supply the size of the cycles to be generated. In reality he specifies the size of one half of the cycle, that is only the "high" portion. Two sizes must be specified. The first size in columns 59-60 is the size of the cycle containing target number one and the second size is the size of the cycle containing the last target. One restriction is that the period of the first cycle (in targets) must be less than or equal to the period of the last cycle, thus, the period of the cycles is always increasing. The number of targets in each half of a cycle is the same.

As mentioned before the number of targets specified for the cycle option is only approximation. Using the period (in targets) of the first and last cycle, it is determined how many targets are required for one cycle of each period between these limits. This number of targets is divided into the number of targets specified and if there is any remainder the result is upped to the next interger. Therefore, the actual number of targets used is always greater than or equal to the number specified. With all other pattern options the number of targets specified is the number of targets used.

LINEAR TARGET PATTERN OPTION

The linear pattern option has been included to give the user a set of targets which is suited to testing the linearity of a processor. The linear pattern consists of a set of targets whose reflection coefficient amplitude is linearly increasing or decreasing and whose phase is a single constant or is uniformly distributed on $(-\pi, \pi)$. The option is taken by placing 'LINEAR' in columns 11-16 of the 'TARGET' data card. The amplitude for the first and last targets go in columns 40-45 and 50-55 respecitvely. All targets are constants, there is no statistical option. If a constant phase is desired 'CPHASE' should be put in columns 69-74 and the phase in columns 75-80. Otherwise the phase is uniform on $(-\pi, \pi)$.

POINT TARGET PATTERN OPTION

The point option was included as an analytic tool to allow the user to see the processor response to a point target. The point option consists of a series of targets which are zero in both amplitude and phase except for the very center target (there is always an odd number of targets) which is the point. This point may be constant or statistical in both amplitude and phase. The number of targets on each side of the point is equal to the number of targets in the antenna used to image the point. To take this option 'POINT' must appear in columns 11-15 of the TARGET data card. The user need not specify a particular number of targets as this depends only on the beam size of the physical antenna used (in terms of targets). The point amplitude may be specified as constant by placing 'CONSTANT' in columns 30-37 of the TARGET data card and the constant in columns 40-45. Otherwise, the amplitude is a number picked from a Rayleigh distribution times the user supplied constant in columns 40-45. The phase may be specified as a constant by placing 'CPHASE' in columns 69-74 and the constant for the phase in columns 75-80. Otherwise, the phase is picked from a uniform distribution in the internal $(-\pi, \pi)$.

EXTERNAL PATTERN OPTION

The purpose of the external target pattern option is to give the user flexibility in inputting his targets for the program. This option involves writing a subroutine named EXTARG which can be called by the program to generate the targets. Thus through this subroutine the user may calculate his own targets or input them via cards or tape.

EXTARG is called whenever an error free data set containing 'EXTARG' in columns 11-16 of the 'TARGET' data card is read in. If an error is detected in the labeling of the cards or in the data on the cards, the subroutine is not called. At present a dummy subroutine EXTARG is included in the deck. When the user writes his own subroutine the must remove this dummy. However, certain cards are required in the user written routine. These are:

```
SUBROUTINE EXTARG(CELPHA,CELAMP,L,K)
DIMENSION CELPHA(L),CELAMP(L)
COMMON/PRO2/FIL1(5),KSAMP1,FI12(14)
.
.
.
RETURN
END
```

The labeled common is necessary only when the user wishes to increase the sampling rate for the statistical analysis of his targets which is KSAMP1. This is one unless changed. The first argument is the target phase array to be created. The second is the target amplitude array to be created. The third is the number of targets to be generated. This is the number input in columns 21-25 of TARGET data card and it must not be changed in this routine. The fourth argument is used to tell the difference between the dummy routine and the valid user written routine. On exist from the user routine K must equal four.

If the user inputs his targets by cards these cards should come right after the last program data card for that set.

RECEIVER DATA CARD

The RECEIVER data card specifies the characteristics of the receiver used to amplify the return signal, including any system noise that may be present. This system noise is narrow band Gaussian additive noise and is added to the signal before it is amplified. The user must input two values to specify the amount of noise to be added. There are two cases. The first, any pattern selection except EXTARG, the user must input a desired signal to noise ratio in columns 36-45. If this value is 999.99 no noise will be added and the second input is not required. Otherwise, he must also input a mean voltage reflection coefficient amplitude on which to base this signal to noise ratio in columns 46-55. If the EXTARG pattern option is taken the user must supply a signal to noise ratio as in the first case and also the average return amplitude in columns 56-65.

Once the noise has been dealt with the return amplitude is amplified according to the following formula.

$$V_{IN} < D \qquad V_{OUT} = V_{IN} \cdot K$$

$$V_{IN} > D \qquad V_{OUT} = K \cdot D + K \cdot \log_{10}(V_{IN} - D + 1)$$

Where D is the linear saturation point or clipping point and is input in columns 16-25, K is the linear gain and is input in columns 11-15, and A is the logrithm multiplier and is imput in columns 26-35.

For a completely noiseless receiver with no amplification of the signal the following values should be input.

K = 1. D = 999. A = 0 S/N = 999.99

ANTENNA DATA CARD

The ANTENNA data card specifies the characteristics of the antenna which creates the returns that will be images. The user must supply an integer number of resolution cells for the antenna in columns 11-15 an integer number of targets per resolution cell in columns 21-25, and a number in columns 41-45 to indicate the weighting function to be applied to the antenna. A doppler mismatch is specified as a real number in columns 31-35. This number may be zero. The sampling rate for the statistical analysis of the returns is input in columns 51-55. At the present time the number for the weighting function must be 1 or 2, 1 for a square beam with all weights equal to one, 2 for a 'cosine on a pedestal' weighting function. If the user wishes to expand the number of weighting options he should see subroutine WGTFCN.

PROCESSOR DATA CARD

The PROCESSOR data card specifies the type of processing that will be used to image the targets. There are several options that may be takn on this card. Full focused processing is specified by placing 'FULL' in columns 11-14. The alternative to this is zero plate processing in which 'ZONE' appears in that field. The difference between these two is that zone uses the sign of the reference function. Next is the option of using one or two channels of the return during imaging. If a "1" appears in column 20 only the cosine channel of the return is used. If a "2" appears both the sine and cosine channels are used. The size of all sub-apertures in terms of the number of resolution cells must be in columns 31-35. This number must not be greater than the number of resolution cells in the physical antenna beam used to create the returns. Columns 41-45 must contain the number of targets incremented before each return is generated. This is analogous to the PRF. The next field, columns 51-55, contain the overlap, in fully compressed returns, of adjacent sub-apertures. This number may be zero, in which case the apertures just touch; positive, in which case they do overlap; or negative, in which case they are separated by so many returns. The starting point for the first sub-aperture in fully compressed returns must be in columns 61-65. This number must be positive. If it is one the first sub-aperture starts at the left hand edge of the beam. The weighting function to be used on each sub-aperture during the imaging is in columns 71-75. At present this number may only be a "1", all weights are one, or a "2", cosine on a pedestal. The user may increase the number of options available by expanding subroutine WGTFCN. See that subroutine for details.

LABELED COMMON VARIABLES

In the following list only the most commonly used name is listed. If a subroutine does not use this name it is only necessary to check its position in the labeled common to find the name it does use.

	PXCH	
1.	PCNUM1	value input in columns 59-60 of TARGET data card.
2.	PCNUM2	value input in columns 64-64 of TARGET data card.
3.	MAXA	value input in columns 40-45 of TARGET data card.
4.	MINA	value input in columns 50-55 of TARGET data card.
5.	LENGTH	number of targets requested
6.	SECON	tells if it is to be constant or statistical amplitude for targets
7.	CHANGE	tells if it is to be constant or statistical phase for targets
8.	AG	constant value for phase
9.	NTIMES	cycle pattern only, number of cycles of each size

	SIGCOM	
1.	KAN1	number of resolution cells in physical beam of antenna.
2.	KANTEL	number of targets in physical beam of antenna
3.	KAN2	number of targets per resolution cell
4.	NP	length of each subaperture in fully compressed returns.

	CELENG	
1.	TLONG	number of targets being used
2.	NUMA	number of returns fully compressed
3.	NUMB	number of returns compressed by INCR only
4.	IA	pointer to CELPHA
5.	IB	pointer to CELAMP
6.	IC	pointer to ANTPHA
7.	ID	pointer to REAMP
8.	IE	pointer to REPHA

9.	IF	pointer to REAMPS
10.	IG	pointer to REAMPC
11.	IH	pointer to WGT
12.	II	pointer to SIGANG
13.	IJ	pointer to COSANG
14.	IK	pointer to SPROD
15.	IL	pointer to CPROD
16.	IM	pointer to AMAPS
17.	IN	pointer to AMAP
18.-29.	INUMB(12)	not used
	RECEIV	
1.	GAIN	linear gain of receiver
2.	SAT	saturation or clipping point of receiver
3.	ALGAIN	logrithmic multiplier to signal above clipping point
4.	STN	signal to noise ratio
	PRO 2	
1.	KFIRST	starting point of first subaperture in fully compressed returns.
2.	KPROC	type of processing 'FULL', 'ZONE', or 'NONE'.
3.	KSAMPB	sampling rate for analysis of images
4.	LENREF	length of fully compressed reference function
5.	LENUNC	length of semi-compressed (INCR only) reference function
6.	KSAMP1	sampling rate for analysis of targets
7.	KSAMP2	sampling rate for analysis of returns
8.	LOOPY	= 1 if statistical analysis is to be perfromed
9. -19.	IFIL (11)	not used
20.	L4	number of images to be produced.

	OPTION	
1.	KPRESM	number of returns to be summed before imaging
2.	IQUANT	number of bits available to store returns
3.	RANG1	lower limit of quantization range
4.	RANG2	upper limit of quantization range
5.	PLOT	= 'YES' if plot option is taken
6.	PUNCH	= 'YES ' if final image is to be punched
7.	SELECT	target pattern selection
8.	FRACBW	doppler mismatch reference to return signal band width
9.	NUMPLT	= 'ALL' plot all subapertures otherwise only final result
10.	KQUANT	number of bits available to store weighted reference function
11.	KPAT	points to subaperture weighting function
12.	KUMP	='YES' if dump option is taken
13.	NCHAN	number of channels to be used in imaging
14.	PUNTAR	= 'YES' if targets amplitudes are to be punched
15.	RCA	mean voltage reflection coefficient amplitude on which the signal to noise ratio is based.
16.	AVEAMP	average return amplitude used to add noise to external targets
17.-20.	FIL(4)	not used
	APERTU	
1.-100.	APWGTS(100)	weight for each subaperture read from APWGTS data card
	ISPACZ	
1.	MAX	amount of blank common available for current run.
2.-4.	FIL(3)	not used
5.	NUMB2	amount of surplus blank common available for current data set

GENERAL FLOW OF CONTROL

This section gives a general idea as to the sequence of events during execution of the program, including the calls to subroutines and what the purpose of the call is.

Execution starts in the mainline which calls ISPACE with argument of zero. This returns the amount of blank common available during this particular run and sets the core hole so that none of this blank common is used for input-output buffers. It is in this blank common region that all dynamic arrays will be created. The mainline next calls subroutine SETWRP which sets an address in word 23 so that control is transferred to subroutine TWRAP in case of an abort, such as an I8 (run time exhausted). This wrap-up routine insures that all data sets written on the tape during the current run are not lost. These two calls are made only once during a particular running of the program. Once these preliminary steps are taken care of, SETARG is called to start processing the first data set. SETARG will return ten arrays in the blank common region along with their pointers and several miscelleneous variables in labeled common. The first step taken by SETARG is to call INSPEC which will input a set of ten data cards. These data cards fully specify the type of targets and imaging to be carried out by the rest of the program. After inputting the cards INSPEC checks the data for twenty-three error conditions and will abort that set with an error message if any one of them is present and immediately begin inputting the next set. If the set specifies that it is to be put on tape TREADY is called to position the tape so that this set may be added. If a card with blanks in columns 1-6 is detected anywhere in the data, control is transferred to the wrap-up routine TWRAP to copy the data back to the tape, if necessary, and upon return from it execution is terminated.

Once a correct data set has been input, control is returned to SETARG which tests to determine which target pattern was selected then transfers control to handle that pattern. All patterns except the TAPE pattern are handled very much alike. NGET is called to calculate the lengths of several arrays which will be used and then tested to see if this data set will fit in the available memory. If not, an error return is taken which by-passes calculating the targets and returns. The heading is output followed by an error message stating that the memory limits must be increased to run this data set. If the set will fit NGET calculates the pointers for all the arrays to be used in SETARG. Upon a normal return form NGET, SETARG calls the appropriate subroutine to calculate the type of targets called for; STAR2 for the cycle pattern, BIGPNT for the point pattern, LINEAR for the linear pattern, and EXTARG for the external pattern option. Each of these generates its particular type of target and then returns to SETARG. ANTENA is called next which simulates a radar antenna. WGTFCN is called to provide an antenna weighting function and an antenna phase array is generated which depends on the size and number of resolution cells in the physical beam of the antenna. Using these and the original targets a weighted sum is performed for certain positions of the antenna beam over the targets, these positions depend on the number of targets skipped per return which is an input to the program. The returns that are generated include the sine and cosine channels and a combination of these which give the return amplitude and return phase. ANTENA also generates the sine and cosine reference functions which will be used in imaging. Back in SETARG, TWRITE is called to write the set on tape if that is specified. If the tape pattern was originally specified, TREADY is called with the data set number as an argument. TREADY searches the tape for the correct data set and reads in the first part of it. NTGET is called to see if the set will fit in the available memory, if not it is aborted. Otherwise, the second part which contains the arrays is read in. CALREF is called to calculate the reference functions to be used in imaging and then control is returned to SETARG. The first part of the heading dealing with the targets is then output and OUT2 is called to output the rest of the heading. Control then returns to the mainline.

SNAP is called from the mainline at this point to print out the target reflection coefficient amplitudes. MENVAR is called to statistically analyze these target amplitudes if required. The targets are next punched on cards if the punch option has been taken. If the dump option has been taken the antenna phase and antenna weighting function are printed here as they are destroyed later. NOISY is then called to simulate a radar receiver. Noise is added according to an input signal to noise ratio unless this number is 999.99 in which case no noise is added. Linear gain and logrithmic gain are next multiplied by the signal. These returns are then analyzed by a call to MENVAR, if indicated. If presumming is indicated, PRESUM is called to sum the desired number of elements of the sine and cosine channels of the returns. Next if quantization is indicated QUANT is called treating the sine and cosine channels as one long array to be quantized. DIGPRO, the digital processor, is called next to image the returns unless the no imaging option is taken in which case the call to DIGPRO is skipped. The first step taken by DIGPRO is to call WGTFCN which generates a weighting function which will be applied to each sub-aperture used. This weighting function is generated based on the length of the sub-aperture in targets. It is then compressed in DIGPRO by the targets incremented per return in generating the returns times the number of returns presummed so as to match up correctly with the returns to be imaged. It is also normalized to the interval (0.,+1.). If Zone processing is indicated the elements of the sine and cosine reference functions are transformed into a plus or minus one depending on the sign of the element. The actual imaging is done in a large DO loop which images one aperture at a time. The weighted sum of all of these apertures is the final image. The weights for this sum are input on the APWGTS data card. Provision is also made in this loop for quantizing the weighted reference function according to a certain number of bits. No range need be supplied as it is -1 to +1. Apertures are imaged in order from left to right as they appear in the physical beam of the antenna. If an aperture position required is not possible a message is printed and imaging stops at that point. After each aperture has been imaged the results from that aperture are printed out, if statistical analysis is indicated it is performed, and if a plot is called for it is made. When all apertures have been imaged the above sequence is repeated for the final image unless only one aperture was used. The final image is then punched on cards, if that punch option has been taken. Control is then transferred back to the mainline which calls CORDMP, if the dump option has been taken. A transfer is then made back to call SETARG which will begin the process over again for the next data set.

232


```
            INTEGER TLONG
            COMMON CELL(1)
            COMMON/ISPACZ/MAX,NNU,NTL,NKA,NUMB2
            COMMON/PROCES/NAPS,KANT,INCR,LAP
            COMMON/CELENG/TLONG,NUMA,NUMB,IA,IB,IC,ID,IE,IF,IG,IH,II,IJ,IK,IL,
           2      IM,IN,IO,IP,IQ,IR,IS,IT,IU,INUMB(5)
            COMMON/SIGCOM/ KAN1,KANTEL,KAN2,NREAP
            COMMON/RECEIV/GAIN,SAT,ALGAIN,STN                              E
            COMMON/PRO2/KFIRST,KPROC,KSAMPD,LENREF,LENUNC,K1SAMP,K2SAMP,LOOPY,
           2    IFIL(11),L4
            COMMON/OPTION/KPRESM,IQUANT,R9(9),DUMP,NCHAN,PUNTAR,FIL(6)
            DATA YES/3HYES/
            DATA NONE/4HNONE/
      *              ISPACE PUTS AMOUNT OF BLANK COMMON IN MAX
            CALL ISPACE(0)
            WRITE(6,35) MAX                                                     2
         35 FORMAT(67H MAXIMUM NUMBER OF MEMORY LOCATIONS AVAILABLE FOR DYNAMI  5
           2C ARRAYS = ,I5)
      *              SETWRP SETS WRAPUP ADDRESS IN CASE OF AN ABORT
            CALL SETWRP                                                         5
      *              SETARG GENERATES TARGETS AND RETURNS
        100 CALL SETARG                                                         6
            WRITE(6,40)                                                         7
         40 FORMAT(1H1)                                                         9
            CALL SNAP(CELL(IB),16HTARGET AMPLITUDE,TLONG)                       9
      *              PERFORM STATISTICAL ANALYSIS ON TARGETS
            IF(LOOPY.EQ.1) CALL MENVAR(K1SAMP,1,CELL(IB),'TARGETS',TLONG)      10
      *              PUNCH TARGET AMPLITUDES IF REQUIRED
            IT = IB + TLONG - 1                                                13
            IF(PUNTAR.EQ.YES) WRITE(43,143) TLONG,(CELL(I),I=IB,IT)            14
        143 FORMAT (I10,7F10.3/(8F10.3))                                       22
            IF(DUMP.NE.YES) GO TO 15                                           22
            CALL SNAP(CELL(IC),13HANTENNA PHASE,KANTEL)                        25
            CALL SNAP(CELL(IH),15HANTENNA WGT FCN,KANTEL)                      26
      *              NO NOISE IS ADDED IF STN = 999.99
            IF(STN.NE.999.99) CALL SNAP(CELL(ID),16HNOISE FREE REAMP, NUMB)    27
      *              NOISY ADDS SYSTEM NOISE, LINEAR GAIN, AND LOGRITHMIC GAIN
         15 CALL NOISY                                                         30
           2    (CELL(ID),CELL(IE),CELL(IF),CELL(IG),CELL(IC),CELL(IH),NUMB,    E
           3    KANTEL,CELL(IA))                                                E
      *PRESUM RETURNS IF SPECIFIED ON PRESUM DATA CARD
            IF(KPRESM.GE.2) CALL PRESUM(CELL(IF),KPRESM)                       31
      *QUANTIZE RETURNS IF SPECIFIED ON QUANTIZE DATA CARD
      *              NO QUANTIZATION TAKES PLACE IF IQUANT = 36
            IF(IQUANT.GT.0 .AND. IQUANT.LT.36) CALL QUANT(CELL(IF),2*NUMA)     34
      * IA IS POINTER TO CELPHA, LENGTH TLONG
      * IB IS POINTER TO CELAMP, LENGTH TLONG
      * IC IS POINTER TO ANTPHA, LENGTH KANTEL
      * ID IS POINTER TO REAMP, LENGTH NUMB
      * IE IS POINTER TO REPHA, LENGTH NUMB
      * IF IS POINTER TO HEAMPS, LENGTH NUMA
```

```
      * IG IS POINTER TO REAMPC, LENGTH NUMA
      * IH IS POINTER TO WGT2, LENGTH NREAP                             A
      * II IS POINTER TO SINANG, LENGTH LENREF
      * IJ IS POINTER TO COSANG, LENGTH LENREF
      * IK IS POINTER TO SPROD, LENGTH NREAP                            E
            IK = IH + NREAP                                                   37
      * IL IS POINTER TO CPROD, LENGTH NREAP                            E
            IL = IK + NREAP                                                   38
      * IM IS POINTER TO AMARS, LENGTH L4                               E
            IM = IL + NREAP                                                   39
      * IN IS POINTER TO ARRAY AMAP, LENGTH L4                          E
            IN = IM + L4                                                E     40
            IF(KPROC.EQ.NONE) GO TO 101                                       41
      *              DIGPRO IS THE DIGITAL PROCESSOR, IT IMAGES THE RETURNS
            CALL DIGPRO(CELL(II),CELL(IJ),CELL(IA),CELL(IB),CELL(ID),         44
           2  CELL(IE),CELL(IF),CELL(IG),CELL(IN),CELL(IM),CELL(IK),CELL(IL), E
           3  CELL(IH),NUMA,NREAP,L4,TLONG,LENREF,NUMB)
      *              CORDMP OUTPUTS ALL ARRAYS AS THEY ARE AT END OF PROCESSING
      *              IF DUMP OPTION IS TAKEN, CORDMP OUTPUTS ALL CORE AS IT IS
      *              AT THE END OF PROCESSING
        101 IF(DUMP.EQ.YES) CALL CORDMP                                       45
            GO TO 100                                                         48
            END                                                               49
```

23757 WORDS OF MEMORY USED BY THIS COMPILATION

```
      SUBROUTINE SETARG
*              MAIN LINK OF MODULE TO CREAT TEN ARRAYS IN BLANK COMMON
*              USING DYNAMIC ALLOCATION.  THE ARRAYS CREATED ARE IN THE
*              FOLLOWING ORDER--SINE REFERENCE FUNCTION (SINANG), COSINE
*              REFERENCE FUNCTION (COSANG), TARGET PHASE (CELPHA), TARGET
*              AMPLITUDE (CELAMP), RETURN AMPLITUDE (REAMP), RETURN
*              PHASE (REPHA), SINE CHANNEL OF RETURN (REAMPS), COSINE
*              CHANNEL OF RETURN (REAMPC), ANTENNA PHASE (ANTPHA), AND
*              THE ANTENNA WEIGHTING FUNCTION (WGT)
*              COMPONENTS OF MODULE--SUBROUTINES SETARG,INSPEC,NGET,
*              LINEAR,STAR2(BIGPNT),EXTARG,ANTENA,WGTFCN, AND TREADY
*              (WITH ENTRY POINTS TREAD,TWRITE, AND TWRAP)
*              SETARG PROCESSES ONE ACCEPTABLE DATA SET PER CALL
*              STOP CONDITION--BLANK 'TARGET' DATA CARD
      REAL MAXA,MINA,AG
      INTEGER PCNUM1,PCNUM2,TLONG,LENGTH,KANTEL,NARRAY,NUMB,NUMB2,IA,IB,
     1INUMB,ISUM
*     ALPHANUMERIC INPUT  SELECT,SECON,CANGLE
      COMMON CELL(1)
      COMMON/ISPACZ/ MAX,NNU,NTL,NKA,NUMB2
      COMMON /PXCH/ PCNUM1,PCNUM2,MAXA,MINA,LENGTH,SECON,CANGLE,AG ,
     2              NTIMES
      COMMON/CELENG/TLONG,K27    ,NUMB,IA,IB,IC,ID,IE,IF,IG,IH,II,IJ,
     2            INUMB(16)
      COMMON/SIGCOM/ KAN1,KANTEL,KAN2,NP
      COMMON/OPTION/H1(6),SELECT,H2(13)
      DATA PNT,STND,XULL,RLIN/6HPOINT ,6HCYCLE ,6H       ,6HLINEAR/
      DATA TAPE /4HTAPE/
      DATA ST,EX/ 6HDELETE,6HEXTARG/
   33 WRITE(6,250)
  250 FORMAT(1H1)                                                          3
*              SUBROUTINE INSPEC INPUTS ALL PRIMARY DATA CARDS TO THE PROGRAM
      CALL INSPEC(TS,IPAT)                                                 3
*              TRANSFER TO GENERATE TYPE OF TARGETS SPECIFIED BY VALUE OF SELECT
*TRANSFER TO GENERATE TYPE OF TARGETS SPECIFIED BY VALUE OF SELECT
      IF(SELECT,EQ.TAPE) CALL TREAD(KNUM,IPAT,LENGTH,$126,$33)             4
      IF(SELECT,EQ.STND) GO TO 71                                          7
      IF(SELECT,EQ.PNT) GO TO 72                                          10
      IF(SELECT,EQ.RLIN) GO TO 73                                         13
      IF(SELECT,EQ.EX) GO TO 74                                           16
      WRITE(6,25)SELECT,TS,LENGTH,KANTEL,SECON,MAXA,MINA,PCNUM1,PCNUM2,   19
     1CANGLE,AG
   25 FORMAT(1H0,A6,1X,A1,2X,I5,5X,I5,4X,A6,2(4X,F5.2),2(3X,I2),3X,A6,    22
     2F6.2,//
     259H NO ACCEPTABLE PATTERN SELECTION, THIS DATA CARD IS SKIPPED  )
      GO TO 33                                                            22
*              START OF EXTERNAL TARGET PROCESSING
   74 NUMB = LENGTH + KANTEL - 1                                          23
      KNUM = 5                                                            24
      TLONG = LENGTH                                                      25
      CALL NGET(NUMB,TLONG,KANTEL,$126)                                   26
```

```
          CALL EXTARG(CELL(IA),CELL(IB),LENGTH,KNUM)                    27
          IF(KNUM.EQ.5) GO TO 126                                       28
          GO TO 1250                                                    31
    *               START OF CYCLE PROCESSING
     71   ISUM=0                                                        32
          KNUM = 3                                                      33
    *               PCNUM1 = ONE HALF MINIMUM NUMBER OF TARGETS PER CYCLE
    *               PCNUM2 = ONE HALF THE MAXIMUM
          DO 88 J = PCNUM1,PCNUM2                                       34
          ISUM=ISUM+2*J                                                 35
     88   CONTINUE                                                      36
    *               ISUM IS NOW NUMBER OF TARGETS REQUIRED FOR ONE CYCLE OF EACH
    *               SIZE PCNUM1*2 THRU PCNUM2*2
          IOVER=0                                                       38
          IF((LENGTH/ISUM)*ISUM.NE.LENGTH) IOVER=1                      39
    *               NTIMES = NUMBER OF CYCLE OF EACH SIZE
          NTIMES = LENGTH/ISUM+IOVER                                    42
    *               TLONG = NUMBER OF ACTUAL TARGETS
          TLONG = NTIMES*ISUM                                           43
          NUMB = TLONG + KANTEL -1                                      44
    *               NGET PLACES AMOUNT OF SURPLUS COMMON IN VARIABLE NUMB2
    *               ALSO CALCULATES POINTERS FOR REST OF ARRAYS NEEDED IN SETARG
    *               NONSTANDARD RETURN IF NUMB2 IS NEGATIVE
          CALL NGET (NUMB,TLONG,KANTEL,$126)                            45
    *               STAR2 GENERATES TARGETS FOR CYCLE PATTERN
    172   CALL STAR2 (CELL(IA),CELL(IB),TLONG)                          46
          GO TO 1250                                                    47
    *               START OF POINT PROCESSING
     72   LENGTH = 1                                                    48
          KNUM = 1                                                      49
          TLONG = 2*KANTEL + 1                                          50
          NUMB = TLONG + KANTEL - 1                                     51
          CALL NGET(NUMB,TLONG,KANTEL,$126)                             52
          CALL BIGPNT(CELL(IA),CELL(IB),KANTEL,TLONG)                   53
          GO TO 1250                                                    54
    *               START OF LINEAR
     73   TLONG = LENGTH                                                55
          KNUM = 2                                                      56
          NUMB = TLONG + KANTEL - 1                                     57
          CALL NGET (NUMB,TLONG,KANTEL,$126)                            58
          CALL LINEAR (CELL(IA),CELL(IB),TLONG)                         59
    *               ANTENA GENERATES RETURNS AS SEEN BY A RADAR ANTENNA
    1250  CALL ANTENA(CELL(IA),CELL(IB),CELL(IF),CELL(IG),CELL(IC),CELL(ID),  60
         2      CELL(IE),CELL(IH),KANTEL,NUMB,TLONG,IPAT,CELL(II),CELL(IJ))
    *               OUTPUT HEAD--SUMMARY OF RESULTS OF THIS MODULE
    125   IF(TS.NE.XULL) CALL TWRITE(KNUM,IPAT)                         61
    126   WRITE(6,45) SELECT,NUMB2,LENGTH,TLONG,GANGLE,AG               64
     45   FORMAT(1H ,27HTARGET PATTERN SELECTED =  ,A6,/                67
         138H NUMBER OF SURPLUS MEMORY LOCATIONS =  ,I6/
         131H NUMBER OF TARGETS SPECIFIED = ,I5,/
         233H ACTUAL NUMBER OF TARGETS USED = ,I5,/
         7     102H SPECIFIED TYPE OF DISTRIBUTION FOR PHASE(IF BLANK, ASSU
```

```
      7HED TO BE UNIFORMLY DISTRIBUTED, -PI TO PI) = ,A6/
      933H CONSTANT PHASE(IF APPLICABLE) = ,F6.2)
*         BRANCH TO OUTPUT VALUES NECESSARY TO EACH PATTERN
       GO TO (212,211,210,213,33),KNUM                              67
  210  WRITE(6,46)SECON,MAXA,MINA,PCNUM1,PCNUM2,NTIMES              68
   46  FORMAT (25H   CYCLE PATTERN ONLY--- ,/                       71
      4 97H   SPECIFIED TYPE OF DISTRIBUTION FOR AMPLITUDE (IF BLANK, ASS
      4UMED TO BE RAYLEIGH DISTRIBUTED) = ,A6/
      571H   AMPLITUDE FOR FIRST HALF OF CYCLE(STANDARD DEVIATION OR CONS
      1TANT) = ,F6.2/
      671H   AMPLITUDE FOR LAST HALF OF CYCLE(STANDARD DEVIATION OR CONST
      2ANT) = ,F5.2/
      658H   ONE HALF THE NUMBER OF TARGETS PER CYCLE AT LOW END = ,I2/
      258H   ONE HALF THE NUMBER OF TARGETS PER CYCLE AT HIGH END = ,I2/
      368H   NUMBER OF CYCLES FOR EACH CONSTANT NUMBER OF TARGETS PER CYC
      3LE = ,I2)
       GO TO 213                                                    71
  211  WRITE(6,47) MAXA,MINA                                        72
   47  FORMAT(25H   LINEAR PATTERN ONLY--- ,/                       75
      131H   AMPLITUDE OF FIRST TARGET = ,F6.2/
      231H   AMPLITUDE OF LAST TARGET = ,F6.2)
       GO TO 213                                                    75
  212  WRITE(6,48) SECON,MAXA                                       76
   48  FORMAT (                                                     79
      495H SPECIFIED TYPE OF DISTRIBUTION FOR AMPLITUDE (IF BLANK, ASSUME
      4D TO BE RAYLEIGH DISTRIBUTED) = ,A6/
      5 52H TARGET AMPLITUDE(STANDARD DEVIATION OR CONSTANT) = ,F6.2)
  213  CALL OUT2(IPAT)                                              79
       IF(NUMB2.LT.0) GO TO 50                                      80
       RETURN                                                       83
*         ERROR MESSAGE IF INSUFFICIENT MEMORY
   50  WRITE (6,355)                                                84
  355  FORMAT(//////29H INSUFFICIENT MEMORY SUPPLIED /              86
      225H THIS DATA SET IS SKIPPED  /
      328H RERUN WITH INCREASED LIMITS    )
*         BRANCH TO INPUT NEW DATA SET IF CURRENT SET ABORTED
       GO TO 33                                                     86
       END                                                          87

<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
    TS     DOES NOT APPEAR IN READ, DATA, COMMON OR LEFT OF EQUALS (=)
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>

<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
    IPAT   DOES NOT APPEAR IN READ, DATA, COMMON OR LEFT OF EQUALS (=)
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>

    23672 WORDS OF MEMORY USED BY THIS COMPILATION
```

```
      SUBROUTINE INSPEC (TS,IPAT)
C***AT PRESENT INSPEC INPUTS  10  DATA CARDS
C*THESE CARDS HAVE THESE LABELS STARTING IN COLUMN ONE AND MUST APPEAR
C*IN THIS ORDER--1) TARGET  2) ANTENNA  3) RECEIVER  4) PROCESSOR
C*  5) APWGTS  6) PRESUM  7) QUANTIZE  8) PLOT  9) PUNCH  10) DUMP
      COMMON/PXCH/PCNUM1,PCNUM2,MAXA,MINA,LENGTH,SECON,CANGLE,AG,NTIMES
      COMMON/SIGCOM/KAN1,KANTEL,KAN2,NP
      COMMON/CELENG/LONG,K54,NUMB,INUMB(26)
      COMMON/RECEIV/GAIN,SAT,ALGAIN,STN
      COMMON/PROCES/NAPS,KANT,INCR,LAP
      COMMON/PRO2/KFIRST,KPROC,KSAMPD,LENREF,LENUNC,KSAMP1,KSAMP2,LOOPY,
     2              IFIL(12)
      COMMON/OPTION/ KPRESM,IQUANT,RANG1,RANG2,PLOT,PUNCH,SELECT,FRACBW,
     2   NUMPLT,KQUANT,KPAT,DUMP,NCHAN,PUNTAR,RCA,AVEAMP,FIL(4)
      COMMON/APERTJ/AWGTS(100)
      DIMENSION TARG(10),ABORT(25),EMESS(250)
C*            ERROR MESSAGES FOR DIAGNOSTICS
      DATA (EMESS(I),I=1,190,10)    /
     1 60HNUMBER OF TARGETS SPECIFIED IS NON-POSITIVE                  ,
     2 60HSAMPLING RATE OF IMAGES DURING STAT. ANALYSIS IS INCORRECT   ,
     3 60HSIZES OF CYCLES FOR CYCLES PATTERN IS INCORRECT              ,
     4 60HSIZE OF ANTENNA (RESOLUTION CELLS) IS NON-POSITIVE           ,
     5 60HSIZE OF RESOLUTION CELL IS NON-POSITIVE                      ,
     6 60HANTENNA WEIGHTING SELECTOR IS NON-POSITIVE                   ,
     7 60HLINEAR GAIN OF RECEIVER IS NEGATIVE                          ,
     8 60HSATURATION POINT OF RECEIVER IS NEGATIVE                     ,
     9 60HLOGRITHMIC GAIN OF RECEIVER IS NEGATIVE                      ,
     1 60HBOTH LINEAR AND LOGRITHMIC GAINS ARE ZERO                    ,
     2 60HSIGNAL TO NOISE RATIO IS NON-POSITIVE                        ,
     3 60HNUMBER OF APERTURES IS NON-POSITIVE                          ,
     4 60HSIZE OF APERTURE IS NOT IN CORRECT RANGE                     ,
     5 60HSAMPLING RATE OF RETURNS (PRF) IS NON-POSITIVE               ,
     5 60HREFERENCE FUNCTION WEIGHTING SELECTOR IS NON-POSITIVE        ,
     6 60HSTARTING POINT OF FIRST APERTURE IS NON-POSITIVE             ,
     6 60HNUMBER OF RETURNS TO BE PRESUMMED IS NON-POSITIVE            ,
     7 60HNUMBER OF BITS TO QUANTIZE RETURNS IS NOT IN CORRECT RANGE   ,
     8 60HLIMITS FOR QUANTIZATION OF RETURNS ARE INCORRECT             /
      DATA EMESS(191)/
     9 60HNUMBER OF BITS TO QUANTIZE REF FCN IS NOT IS CORRECT RANGE   /
      DATA EMESS(201) /
     2 60HALL APERTURE WEIGHTS ARE ZERO                                /
      DATA EMESS(211)/
     2 60HNUMBER OF CHANNELS TO BE USED IN PROCESSING IS INCORRECT     /
      DATA EMESS(221)  /
     2 60HCONSTANT FOR TARGET PHASE IS NOT IN CORRECT RANGE            /
      DATA POINT/5HPOINT/
      DATA(TARG(I),I=1,4)/6HTARGET,6HANTENN,6HRECEIV,6HPROCES/
      DATA A2,A3/6HTAPE  ,6HDELETE/,XULL/6H      /
      DATA (TARG(I),I=5,8)/6HPRESUM,6HQUANTI,6HPLOT  ,6HPUNCH /
      DATA TARG(9),TARG(10)/4HDUMP,6HAPWGTS/
      DATA CYCLE,CPHASE,LABORT /6HCYCLE  ,6HCPHASE,25/
```

```
      * READ TARGET SPECIFICATION CARD
  133 READ(5,10) CARD,SELECT,TS,INDIC,LENGTH,SECON,MAXA,KSAMPD,MINA,
     2 PCNUM1,PCNUM2,CANGLE,AG
   10 FORMAT(A6,4X,A6,1X,A1,I1,1X,I5,4X,A6,4X,F6.2,1X,I2,1X,F6.2,2(3X,      9
     2 I2),3X,A5,F6.2)
      K = 1                                                                 9
*                 IS THIS THE RIGHT CARD
   34 IF(CARD.EQ.TARG(K)) GO TO (1,2,3,9,5,6,7,8,666,4),K                  10
*                 ARE WE DONE
  111 IF(CARD.NE.XULL) GO TO 11                                            13
   14 CALL THRAH                                                           16
      CALL EXIT                                                            17
   11 BACKSPACE 05                                                         18
*                 WHEN SEARCHING FOR TARGET CARD INPUT CARDS UNDER A FORMAT
*                 TO AVOID BAD FORMAT ERROR MESSAGES
   13 READ(5,201) CARD,(ABORT(I),I=1,13)                                   19
  201 FORMAT(13A6,A2)                                                      25
      IF(CARD.NE.TARG(1)) GO TO 12                                         25
      BACKSPACE 05                                                         28
      GO TO 133                                                            29
   12 IF(CARD.EQ.XULL) GO TO 14                                            30
      WRITE(6,20)                                                          33
   20 FORMAT(46H DATA CARDS OUT OF ORDER, THIS CARD IS SKIPPED   /         35
     2 25H SEARCH FOR TARGET CARD  )
      WRITE(6,201) CARD,(ABORT(I),I=1,13)                                  35
      GO TO 13                                                             41
*READ 'ANTENNA' DATA CARD
    1 READ(5,30) CARD,KAN1,KAN2,FRACBW,IPAT,KSAMP2                         42
   30 FORMAT(A6,4X,2(I5,5X),F5.2,5X,I5,5X,I5)                              48
*KAN1 = NUMBER OF RESOLUTION CELLS IN BEAM
*KAN2 = NUMBER OF TARGETS PER RESOLUTION CELL
*IPAT = POINTER TO TYPE OF WEIGHTING FOR ANTENNA
*                FRACBW = DOPPLER MISMATCH REFERENCED TO RETURN SIGNAL BAND WIDTH
*                KSAMP2 = SAMPLING RATE FOR ANALYSIS OF RETURNS
      K = 2                                                                48
      GO TO 34                                                             49
    2 READ(5,40) CARD,GAIN,SAT,ALGAIN,STN,RCA,AVEAMP                       50
      K = 3                                                                53
   40 FORMAT(A6,4X,F5.2,5F10.2)                                            54
*GAIN = LINEAR GAIN OF RECEIVER
*SAT = SATURATION OR CLIPPING POINT OF RECEIVER
*ALGAIN = LOGRITHMIC GAIN OF RECEIVER
*                STN = SIGNAL TO NOISE RATIO FOR RECEIVER
*                RCA = MEAN VOLTAGE REFLECTION COEFFICIENT AMPLITUDE
*                AVEAMP = AVERAGE RETURN AMPLITUDE FOR EXTERNAL TARGETS
      GO TO 34                                                             54
*READ 'PROCESSOR' DATA CARD
    3 READ (5,50) CARD,KPROC,NCHAN,NAPS,KANT,INCR,LAP,KFIRST,KPAT          55
   50 FORMAT(A6,4X,A6,3X,I1,3X,I2,5(5X,I5))                                65
*KPROC = DETERMINES IF PROCESSING WILL BE 'FULL' FOCUS OR 'SEMI' FOCUS
*KFIRST = STARTING POINT OF FIRST SUBAPERTURE IN FULLY COMPRESSED RETURNS
*NAPS = NUMBER OF APERTURES IN BEAM
```

```
      *RANT = NUMBER OF RESOLUTION CELLS PER APERTURE
      *INCR = SAMPLING RATE OF REAMPS(C)
      *LAP = NUMBER OF FULLY COMPRESSED RETURNE OVERLAPPED IN ADJACENT APERTURES
            K = 4                                                          65
            GO TO 34                                                       66
      * READ 'APWGTS' DATA CARD
         9  READ(5,90) CARD,(AWGTS(I),I=1,NAPS)                            67
        90  FORMAT(A6,4X,/F10.2/(8F10.2))                                  73
            K = 10                                                         73
            GO TO 34                                                       74
      * READ  'PRESUM' OPTION CARD
         4  READ(5,60) CARD,KPRESM                                         75
        60  FORMAT(A6,4X,I5)                                               78
      * KPRESH = NUMBER OF REAMPS TO BE SUMMED BEFORE PROCESSING
            K = 5                                                          78
            GO TO 34                                                       79
      *NOT TAKEN
      *READ 'QUANTIZE' DATA CARD
         5  READ(5,70) CARD,IQJANT,RANG1,RANG2,KQUANT                      80
        70  FORMAT(A5,4X,I5,2(5X,F5.2),5X,I5)                              84
      *                KQUANT = NUMBER OF BITS AVAILABLE TO STORE REF FCN
      *IQUANT = NUMBER OF BITS AVAILABLE FOR WORD DURING PROCESSING
      *NUMBER OF QUANTIZATION LEVELS = 2**IQUANT
            K = 6                                                          84
            GO TO 34                                                       85
         6  READ(5,80) CARD,PLOT,NUMPLT                                    86
        80  FORMAT(A6,4X,A5,4X,A6)                                         89
            K = 7                                                          89
            GO TO 34                                                       90
         7  READ(5,80) CARD,PUNCH,PUNTAR                                   91
            K = 8                                                          94
            GO TO 34                                                       95
         8  READ(5,80) CARD,DUMP                                           96
            K = 9                                                          99
            GO TO 34                                                      100
       666  DO 99 I=1,LABORT                                              101
            ABORT(I) = 0.                                                 102
        99  CONTINUE                                                      103
      *                CHECK DATA SET FOR ERRORS
      *                IF DATA SET IS ON TAPE SKIP PART OF ERROR CHECKING
            IF(SELECT,EQ.A2) GO TO 81                                     105
            IF(LENGTH,LE.0 .AND. SELECT.NE.POINT) ABORT(1) = 1.           108
            IF(MAXA.EQ.MINA .AND. SELECT.EQ.CYCLE .AND. KSAMPD.LE.0)ABORT(2)=1   111
            IF(PCNUM1,GT.PCNUM2 .AND. SELECT.EQ.CYCLE) ABORT(3) = 1.      114
            IF(KAN1,LE.0)                        ABORT(4) = 1.            117
            IF(KAN2,LE.0)                        ABORT(5) = 1.            120
            IF(IRAT,LE.0)                        ABORT(6) = 1.            123
            IF(KANT.LE.0 .OR. KANT.GT.KAN1)      ABORT(13) = 1.           126
            IF(CANGLE,EQ.CPHASE .AND. (AG.GT. 3.14159 .OR. AG.LT.(-3.14159))  129
         2  )                                    ABORT(23) = 1.
            IF(INCR.LE.0)                        ABORT(14) = 1.           132
        81  IF(GAIN,LT.0.)                       ABORT(7) = 1.            135
```

```
              IF(SAT.LT.0.)                                ABORT(8) = 1.          138
              IF(ALGAIN.LT.0.)                             ABORT(9) = 1.          141
              IF(GAIN.EQ.0. .AND. ALGAIN.EQ.0.)            ABORT(10) = 1.         144
              IF(STN.LE.0.)                                ABORT(11) = 1.         147
              IF(NAPS.GT.0) GO TO 82                                              150
              ABORT(12) = 1.                                                      153
              GO TO 84                                                            154
         82   DO 187 I = 1,NAPS                                                   155
              IF(AWGTS(I).NE.0.) GO TO 84                                         156
        187   CONTINUE                                                            159
              ABORT(21) = 1.                                                      161
         84   IF(KFIRST.LE.0)                              ABORT(15) = 1.         162
              IF(KPAT.LE.0)                                ABORT(16) = 1.         165
              IF(KPRESM.LE.0)                              ABORT(17) = 1.         168
              IF(IQUANT.LE.0 .OR. IQUANT.GT.36)            ABORT(18) = 1.         171
              IF(RANG1.LT.0. .OR. RANG2.LT.0. .OR. RANG1.GT.RANG2)ABORT(19) = 1.  174
              IF(KQUANT.LE.0 .OR. KQUANT.GT.36)            ABORT(20) = 1.         177
              IF(NCHAN.NE.1 .AND. NCHAN.NE.2)              ABORT(22) = 1.         180
              IF(TS.NE.XULL .AND. INDIC.EQ.1) CALL TREADY(INDIC)                  183
              A = 0.                                                              186
              DO 98  I=1,LABORT                                                   187
         98   A = A + ABORT(I)                                                    188
              IF(A.NE.0.) GO TO 96                                                190
              KANTEL = KAN1*KAN2                                                  193
*                  KANTEL = NUMBER OF TARGETS IN BEAM
              KSAMP1 = 1                                                          194
              IF(TS.NE.XULL .AND. INDIC.NE.1) CALL TREADY (INDIC)                 195
              RETURN                                                              198
         96   WRITE(6,10) TARG(1),SELECT,TS,INDIC,LENGTH,SECON,MAXA,KSAMPD,MINA,  199
             2      PCNUM1,PCNUM2,CANGLE,AG
              WRITE(6,30) TARG(2),KAN1,KAN2,FRACBW,IPAT,KSAMP2                    202
              WRITE(6,40) TARG(3),GAIN,SAT,ALGAIN,STN,RCA,AVEAMP                  205
              WRITE(6,50) TARG(4),KPROC,NCHAN,NAPS,KANT,INCR,LAP,KFIRST,KPAT      208
              WRITE(6,90) TARG(10),(AWGTS(I),I=1,NAPS)                            211
              WRITE(6,60) TARG(5),KPRESM                                          217
              WRITE(6,70) TARG(6),IQUANT,RANG1,RANG2,KQUANT                       220
              WRITE(6,80) TARG(7),PLOT,NUMPLT                                     223
              WRITE(6,80) TARG(8),PUNCH,PUNTAR                                    226
              WRITE(6,80) TARG(9),DUMP                                            229
              WRITE(6,699) A                                                      232
*                  DIAGNOSTICS IF DATA IS UNACCEPTABLE
        699   FORMAT(//1X,F5.0,61H FATAL ERRORS WERE DETECTED IN THE ABOVE DATA   235
             2SET, THEY ARE--   //)
              DO 88 I=1,LABORT                                                    235
              IF (ABORT(I).EQ.0.) GO TO 88                                        236
              J2 = 10*I                                                           239
              J1 = J2 - 9                                                         240
              WRITE (6,188) I,(EMESS(K),K=J1,J2)                                  241
        188   FORMAT(8H ERROR # ,I3,4H -- , 10A6)                                 247
         88   CONTINUE                                                            247
              WRITE(6,189)                                                        249
        189   FORMAT(1H1)                                                         251
```

```
      208          GO TO 133                                          251
      209          END                                                252

    23761 WORDS OF MEMORY USED BY THIS COMPILATION
```

```
      SUBROUTINE TREADY (IN)
*              ENTRY AT TREADY POSITIONS THE TAPE OR DISC SO THAT IT MAY
*              BE WRITTEN ON.  THIS MAY INVOLVE COPYING FROM ONE FILE TO
*              THE OTHER OR NO ACTION AT ALL.
*              IN = 1  IF THIS IS TO INITIALIZE THE TAPE, OTHERWISE IT
*              IS PUT AT END OF THOSE ALREADY ON THE TAPE
      INTEGER TLONG,RT,WT,M(2)
      COMMON/CELENG/TLONG,K54,NUMB,IA,IB,IE,ID,IS,IF,IG,IH,II,IJ,INM(16)
      COMMON/ISPACZ/MAX,NNU,NTL,NKB,NUMB2
      COMMON/PXCH/PCNUM1,PCNUM2,MAXA,MINA,LENGTH,SECON,CANGLE,AG,NTIMES
      COMMON/SIGCOM/KAN1,KANTEL,KAN2,NP
      COMMON/PROCES/NAPS,KANT,INCR,LAP
      COMMON/OPTION/A(5),SELECT,B(43)
      COMMON ICELL(25)
      DATA RT,WT,FIRST,WI,NUM,ON,OFF/2,1,0.,0.,0,1.,0./
      DATA CW,NIL/1.,19/
      DATA MAXSET/100000/
* INITIALLY WRITE INDICATOR, WI, IS OFF
* READ TAPE, RT, IS FILE CODE 2, USER SUPPLIED TAPE
* WRITE TAPE,WT, IS FILE CODE 1, LINKED DISC
*              CW IS INDICATOR, INITIALLY ON, WHICH TELLS IF IT IS POSSIBLE
*              TO ATTEMPT TO WRITE ON THE TAPE.  CW IS SET OFF IF DURING
*              COPYING FROM TAPE TO DISC A DATA SET IS FOUND THAT IS
*              TOO LARGE FOR AVAILABLE CORE AND IS THUS IMPOSSIBLE TO COPY
*              ONCE THE CW INDICATOR IS SET OFF IT STAYS OFF FOR THE REST OF THE
*              RUN AND ANY FUTURE ATTEMPT TO WRITE ON THE TAPE IS IGNORED
*              THE TAPE MAY BE READ AS USUAL, HOWEVER.
      IF(CW.EQ.OFF) RETURN
      ICH = 0                                                      4
      GO TO 10                                                     5
      ENTRY TREAD(K,IPAT,K2,*,*)                                   6
*              ENTRY AT TREAD RESULTS IN READING DATA SET K2 FROM THE
*              TAPE OR AN ERROR MESSAGE STATING THAT IT IS NOT ON THE TAPE
      IF(K2.LE.0 .OR. (WI.EQ.ON .AND.K2.GT.NUM)) GO TO 50          6
      ICH = 1                                                      9
   10 IF(FIRST.NE.0.) GO TO 15                                     10
*              SET FILES FOR EOF RETURN AND REWIND, IF THIS IS THE FIRST
*              ENTRY AT TREADY OR TREAD
      CALL FLGEOF(1,M(1))                                          13
      CALL FLGEOF (2,M(2))                                         14
      REWIND 1                                                     15
      REWIND 2                                                     16
      FIRST = 2.                                                   17
   15 IF(ICH.NE.0) GO TO 20                                        18
      IF(IN.EQ.1) GO TO 24                                         21
*              IF WRITE INDICATOR IS ON, NO ACTION IS REQUIRED
   17 IF(WI.EQ.ON) RETURN                                          24
*              OTHERWISE MUST COPY READ FILE TO WRITE FILE
      REWIND RT                                                    27
      REWIND WT                                                    28
      WI = ON                                                      29
```

```
      30 READ(RT) INUM,(ICELL(I),I=1,N1L)                               30
     *              WHEN HIT EOF WE'RE DONE
         IF(M(RT).EQ.0) GO TO 201                                       36
         M(RT) = 0                                                      39
         RETURN                                                         40
     201 LRS = ICELL(4)                                                 41
         IF(LRS.GT.MAX) GO TO 26                                        42
         NUM = INUM                                                     45
         WRITE(WT)NUM,(ICELL(I),I=1,N1L)                                46
         READ(RT) (ICELL(I),I=1,LRS)                                    52
         WRITE(WT)(ICELL(I),I=1,LRS)                                    57
         GO TO 30                                                       62
      26 CW = OFF                                                       63
         WI = OFF                                                       64
         BACKSPACE RT                                                   65
         LRS = MAX - LRS                                                66
     *              MESSAGE TO TELL DATA SET CAN NOT BE WRITTEN ON TAPE AND
     *              HOW MUCH TO INCREASE LIMITS TO BE ABLE TO WRITE ON THE TAPE
         WRITE(6,79) LRS                                                67
      79 FORMAT(50H IT WAS IMPOSSIBLE TO WRITE THIS DATA SET ON TAPE    /  70
        2 62H TO DO ANY WRITING ON THE TAPE INCREASE THE LIMITS BY AT LEAST
        3     I5, 18H MEMORY LOCATIONS   /
        4 95H ALL FUTURE ATTEMPTS DURING THIS RUN TO WRITE ON THE TAPE ARE
        5IGNORED WITH NO MESSAGE WRITTEN    )
         RETURN                                                         70
     *              HANDLES TAPE INITIALIZATION
      24 IN = 0                                                         71
         IF(FIRST.EQ.2/.) GO TO 81                                      72
         WRITE(6,80)                                                    75
      80 FORMAT(119H ***TAPE INITIALIZATION INDICATED BUT NOT FIRST SET WRI   77
        2TTEN THIS RUN, THEREFORE INITIALIZATION IGNORED***              )
         GO TO 17                                                       77
      81 WI = ON                                                        78
         WT = 2                                                         79
         RT = 1                                                         80
         NUM = 0                                                        81
         REWIND 1                                                       82
         REWIND 2                                                       83
         FIRST = 34.                                                    84
         RETURN                                                         85
         ENTRY TWRITE(K,IPAT)                                           86
     *              WRITES DATA SET ON TAPE ALREADY POSITIONED BY CALL TO TREADY
         IF(CW.EQ.OFF) RETURN                                           86
         NUM = NUM + 1                                                  89
         IH2 = IC - IA + KANTEL                                         90
         WRITE(WT) NUM,TLONG,NUMB,KANTEL,IH2,PCNUM1,                    91
        2   PCNUM2,MAXA,MINA,LENGTH,SECON,CANGLE,AG,NTIMES,KAN1,KAN2,INCR,
        3        K,SELECT,IPAT
         IH3 = IC - 1 + KANTEL                                          94
         WRITE(WT) (ICELL(I),I=IA,IH3)                                  95
         WRITE(6,27) NUM                                               100
      27 FORMAT(39H THIS WAS WRITTEN AS TAPE DATA SET #   , I5)        103
```

244

```
      WI = ON                                                      103
      RETURN                                                       104
* START OF READ PORTION
*                IS FILE READY TO BE READ
   20 IF(WI.EQ.OFF) GO TO 70                                       105
*                IF NOT WRITE EOF ON WRITE FILE AND MAKE IT THE READ FILE
      REWIND WT                                                    108
      MAXSET = NUM                                                 109
      NUM = 0                                                      110
      RT = WT                                                      111
      WT = 3 - RT                                                  112
      WI = OFF                                                     113
   70 IF(MAXSET.LT.K2) GO TO 50                                    114
      IF(3*NUM.GT.4*K2) GO TO 71                                   117
      IBACK = (NUM - K2 + 1)*2                                     120
      IF(IBACK) 21,45,22                                           121
   22 DO 72 I = 1,IBACK                                            122
      BACKSPACE RT                                                 123
   72 CONTINUE                                                     124
      GO TO 45                                                     126
   71 REWIND RT                                                    127
   21 READ(RT) INUM                                                128
*                IF EOF, SET IS NOT ON THE TAPE
      IF(M(RT).NE.0) GO TO 150                                     131
      NUM = INUM                                                   134
      IF(NUM.EQ.K2) GO TO 44                                       135
      READ(RT)                                                     138
      IF(NUM.EQ.(K2-1)) GO TO 45                                   140
      GO TO 21                                                     143
   44 BACKSPACE RT                                                 144
   45 READ(RT) NUM,TLONG,NUMB,KANTEL,IH2,PGNUM1,                   145
     2   PGNUM2,MAXA,MINA,LENGTH,SECON,CANGLE,AG,NTIMES,KAN1,KAN2,
     3   INCR,K,SELECT,IPAT
      IF(M(RT).NE.0) GO TO 150                                     161
      CALL NTGET(NUMB,TLONG,KANTEL,$46)                            164
      IH2 = IA + IH2 - 1                                           165
      READ(RT) (ICELL(I), I=IA,IH2)                                166
      WRITE (6,101) NUM                                            171
  101 FORMAT (29H THIS IS TAPE DATA SET #      ,I3)                174
*                CALREF GENERATES SINE AND COSINE REFERENCE FUNCTIONS
      CALL CALREF(ICELL(IC),ICELL(II),ICELL(IJ),0)                 174
      RETURN 1                                                     175
   46 BACKSPACE RT                                                 176
      NUM = NUM - 1                                                177
      RETURN 1                                                     178
  150 REWIND RT                                                    179
      M(RT) = 0                                                    180
      NUM = 0                                                      181
   50 WRITE(6,47) K2                                               182
   47 FORMAT(10H DATA SET  ,I5,28H DOES NOT APPEAR ON THE TAPE   )  185
      RETURN 2                                                     185
      ENTRY THRAP                                                  186
```

```
      *          WRAP UP ROUTINE TO CLOSE FILES SO NOTHING IS LOST
            IF(FIRST.EQ.0.) RETURN                                                186
      *          WAS ROUTINE ENTERED AT ALL, IF NOT NO ACTION REQUIRED,
      *          OTHERWISE FIND WHERE GOOD INFO IS (TAPE OR DISC) AND LEAVE
      *          IT ON TAPE
            REWIND WT                                                              189
            REWIND RT                                                              190
            IF(WI.EQ.OFF) GO TO 66                                                 191
            IF(WT.EQ.2) GO TO 167                                                  194
            GO TO 65                                                               197
   66       IF(RT.EQ.1) GO TO 65                                                   198
            IF(MAXSET.EQ.100000) GO TO 69                                          201
            NUM = MAXSET                                                           204
            GO TO 67                                                               205
   65       READ(1) INUM,(ICELL(I),I=1,N1L)                                        206
            IF(M(1).NE.0) GO TO 167                                                212
            NUM = INUM                                                             215
            WRITE(2) NUM,(ICELL(I),I=1,N1L)                                        216
            IH2 = ICELL(4)                                                         222
            READ(1) (ICELL(I),I=1,IH2)                                             223
            WRITE(2)(ICELL(I),I=1,IH2)                                             228
            IF (M(1).EQ.0) GO TO 65                                                233
  167       REWIND 2                                                               236
   67       WRITE(6,102) NUM                                                       237
  102       FORMAT(10H DATA SET ,I4,38H IS THE LAST ONE APPEARING ON THE TAPE)     240
   69       FIRST = 0.                                                             240
            RETURN                                                                 241
            END                                                                    242
```

23749 WORDS OF MEMORY USED BY THIS COMPILATION

```
      SUBROUTINE STAR2 (CELPHA.CELAMP,I27)
***FUNCTION--TO PRODUCE TWO ARRAYS, ONE FOR TARGET PHASE AND ONE FOR AMPLITUDE.
*THE GRAPH OF THE AMPLITUDE IS TO BE CYCLES OF HIGHS AND LOWS OF
*INCREASING LENGTH.
*                    ENTRY BIGPNT HANDLES THE POINT PATTERN
      INTEGER CPC1,CPC2
      REAL LOW,CELPHA(I27),CELAMP(I27),HI,ANGL2,URE,FIG
      COMMON /PXCH/CPC1,CPC2,HI,LOW,LONG,S2,CANGL2,ANGL2,NTIMES
*                    CELPHA = ARRAY TO CONTAIN PHASE OF TARGETS
*                    CELAMP = ARRAY TO CONTAIN AMPLITUDE OF TARGETS
*                    I27 = NUMBER OF TARGETS TO BE GENERATED(EQUIVALENT TO TLONG)
      DATA STEST,CONPH/6HCONSTA,6HCPHASE/
      DATA NST,IST/1582634977,825193749/
***K = NUMBER OF TARGET BEING GENERATED
      K=1
***NST - USED IN RANDOM NUMBER GENERATOR
***URE IS ONE IF AMPLITUDE IS CONSTANT, OTHERWISE IS RECALCULATED AS
*  RANDOM NUMBER FOR EACH TARGET
*  SQUARE ROOT OF CHI SQUARED TWO DISTRIBUTION FOR EACH TARGET
*AN3 IS CONSTANT PHASE OR IS RECALCULATED AS RANDOM PHASE FOR EACH TARGET
      AN3=ANGL2                                                              2
*77 LOOP CONTROLS SIZE OF EACH CYCLE
      DO 77 J2=CPC1,CPC2                                                     3
*65 LOOP CONTROLS NUMBER OF CYCLES OF EACH SIZE
      DO 66 JJ=1,NTIMES                                                      4
*FIG IS COMPONENT OF AMPTITUDE, HI ON ONE HALF OF CYCLE, LOW ON OTHER
      FIG=HI                                                                 5
*55 LOOP WHEN TRAVERSED ONCE, GENERATES ONE HALF CYCLE
 155  DO 55 K2=1,J2                                                          6
      IF(S2.NE.STEST) URE=SQRT(RMS(NST)**2+RMS(NST)**2)                      7
      CELAMP(K)=FIG*URE                                                      10
      IF(CONPH.NE.CANGL2) AN3= (RCH(IST)-.5)*6.28                            11
      CELPHA(K)=AN3                                                          14
      K=K+1                                                                  15
      IF(K.GT.I27) GO TO 83                                                  16
  55  CONTINUE                                                               19
      IF(HI.EQ.LOW) GO TO 155                                                21
*CHANGE VALUE OF AMPLITUDE AND REPEAT 55 LOOP
      IF(FIG.EQ.LOW) GO TO 66                                                24
      FIG=LOW                                                                27
      GO TO 155                                                              28
  66  CONTINUE                                                               29
  77  CONTINUE                                                               31
  83  RETURN                                                                 33
      ENTRY BIGPNT (CELPHA.CELAMP,KAN,I27)                                   34
*                    KAN = NUMBER OF TARGETS IN THE ANTENNA
*                    THIS ENTRY POINT GENERATES TARGETS FOR THE POINT PATTERN.  THE
*                    NUMBER OF TARGETS GENERATED DEPENDS ON THE SIZE OF THE ANTENNA
*                    BEING USED.  THE TARGET IS GENERATED SUCH THAT THE FIRST ANTENNA
*                    LENGTH ARE ALL ZEROS THEN THE POINT FOLLOWED BY ANOTHER ANTENNA
*                    LENGTH OF ZEROS
```

```
              DO 199 I=1,127                                        34
              CELPHA(I) = 0.                                        35
              CELAMP(I) = 0.                                        36
        199   CONTINUE                                              37
              K = KAN + 1                                           39
              CELPHA(K) = ANGL2                                     40
              IF(CONPH.NE.CANG_2) CELPHA(K) = (RCM(IST)-.5)*6.28    41
              CELAMP(K) = HI                                        44
              IF(S2.NE.STEST) CELAMP(K)    = SQRT(RMS(NST)**2+RMS(NST)**2)*HI   45
              RETURN                                                48
              END                                                   49

   23671 WORDS OF MEMORY USED BY THIS COMPILATION
```

```
      SUBROUTINE LINSAR(CELPHA,CELAMP,M)
***FUNCTION--TO PRODUCE AS OUTPUT PARMETERS TWO ARRAYS,ONE FOR PHASE AND
*ONE FOR AMPLITUDE.
*GRAPH OF AMPLITUDE IS TO BE LINEAR.  THE ENDPOINTS ARE PATAMETERS THROUGH
*COMMON, FIRST AND LAST.
*               CELPHA = ARRAY FOR PHASE OF TARGETS
*               CELAMP = ARRAY FOR AMPLITUDS OF TARGETS
*               M = NUMBER OF TARGETS USED
      REAL LAST,CELPHA(M),CELAMP(M),FIRST,A47,A47B
      COMMON/PXCH/ FIL1(2),FIRST,LAST,FIL2(2),CAN,A47B,NTIMES
      DATA NST/9578482159/
      DATA CONPHL/6HCPHASE/
      A47 =A47B
*               A IS SLOPE OF PLOT OF AMPLITUDE
      A = (LAST-FIRST)/FLOAT(M-1)                                      2
      CELPHA(1) = A47                                                  3
      IF(CONPHL.NE.CAN) CELPHA(1) = (RCM(NST)-.5)*6.28                 4
      CELAMP(1) = FIRST                                                7
      DO 99 K=2,M                                                      8
      CELAMP(K) = CELAMP(K-1) + A                                      9
      IF(CONPHL.NE.CAN) A47= (RCM(NST)-.5)*6.28                       10
      CELPHA(K)=A47                                                   13
   99 CONTINUE                                                        14
      RETURN                                                          16
      END                                                             17
```

23647 WORDS OF MEMORY USED BY THIS COMPILATION

```
      SUBROUTINE EXTARG(CELPHA,CELAMP,L,K)
      DIMENSION CELPHA(L),CELAMP(L)
*              CELPHA = ARRAY FOR PHASE OF TARGETS
*              CELAMP = ARRAY FOR AMPLITUDE OF TARGETS
*              L = NUMBER OF TARGETS USED,  MUST BE SUPPLIED ON TARGET DATA
*              CARD AND MUST NOT BE CHANGED IN EXTARG
*              K TELLS MAIN PROGRAM IF THIS IS DUMMY PROGRAM (K=5) OR USER
*              WRITTEN PROGRAM (K=4)
      COMMON/PRO2/ FIL1(2),KSAMPD,FIL2(2),KSAMPT,KSAMPR,LOOPY,FIL3(12)
*              KSAMPD = SAMPLING RATE FOR ANALYSIS OF IMAGES
*              KSAMPT = SAMPLING RATE FOR ANALYSIS OF TARGETS
*              KSAMPR = SAMPLING RATE FOR ANALYSIS OF RETURNS
*              LOOPY = 1 WHEN THERE IS TO BE ANALYSIS
*              THE USER MAY SET LOOPY = 1  WHICH WILL RESULT IN STATISTICAL
*              ANALYSIS OF HIS TARGETS, RETURNS AND IMAGES.  SAMPLING RATES
*              FOR RETURNS AND IMAGES WOULD THEN BE REQUIRED AS INPUTS
*              ON THE DATA CARDS.  SAMPLING RATE FOR THE TARGETS IS ONE
*              UNLESS INCREASED BY THE USER IN EXTARG
      K = 5
* DEBUG WRITE STATEMENT
      WRITE (6,10)                                                      2
   10 FORMAT (32H THIS IS DUMMY SUBROUTINE EXTARG)                      4
*USER MUST REMOVE PRECEDING FOUR (4) CARDS WHEN WRITING HIS PROGRAM
***THIS SUBROUTINE IS LEFT OPEN SO THAT THE USER MAY SUPPLY HIS OWN TARGETS
*THE TWO ARRAYS TO BE GENERATED ARE CELPHA FOR THE PHASE OF THE TARGETS
*AND CELAMP FOR THE AMPTITUDE OF THE TARGETS,  L IS THE NUMBER OF TARGETS
*SPECIFIED IN THE FIFTH FIELD OF THE EXTARG TARGET DATA CARD
**THE EXAMPLE BELOW WOULD INPUT FIRST ALL L PHASES AND THEN L AMPLITUDES
*FROM CARDS
C     K = 4
C     READ(5,15) (CELPHA(I),I = 1,L)
C     READ(5,15) (CELAMP(I),I = 1,L)
C 15  FORMAT (10F8.2)
*AT SOME POINT IN THE PROGRAM THE USER MUST SET THE PARAMETER K EQUAL TO 4
*ALSO, THE USER MUST NOT CHANGE THE VALUE OF L (NUMBER OF TARGETS USED)
*FROM THAT WHICH APPEARRED ON THE TARGET DATA CARD
*IF THE USER WISHES TO INPUT TARGETS BY CARDS THROUGH SUBROUTINE EXTARG THESE
*DATA CARDS SHOULD BE PLACED DIRECTLY AFTER THE LAST REQUIRED PROGRAM INPUT CARD
      RETURN                                                            4
      END                                                               5

 23696 WORDS OF MEMORY USED BY THIS COMPILATION
```

```
      SUBROUTINE NGET(NUMB,TLONG,KANTEL,*)
*               NGET DOES THREE BASIC THINGS.  FIRST IT CALCULATES THE LENGTHS
*               OF SEVERAL ARRAYS, SECOND USING THESE AND OTHERS LENGTHS FOR
*               ARRAYS IT DETERMINES THE EXACT AMOUNT OF CORE NECESSARY FOR THIS
*               DATA SET AND IF SUFFICIENT CORE IS AVAILABLE, IF NOT ERROR RETURN
*               WHICH ABORTS CURRENT DATA SET.  IF ENOUGH CORE IS AVAILABLE IT
*               THEN CALCULATES POINTERS TO ALL ARRAYS TO BE USED IN
*               MODULE SETARG
      INTEGER TLONG
      COMMON/ISPACZ/MAX,NNU,NTL,NKA,NUMB2
      COMMON/PROCES/NAPS,KANT,INCR,LAP
      COMMON/SIGCOM/KAN1,KAN3,KAN2,NP
      COMMON/CELENG/ L,NUMA,N45,IA,IB,IC,ID,IE,IF,IG,IH,II,IJ,IK,IL,IM,
     2         IN,IO,IP,INUMB(10)
      COMMON/PXCH/P1,P2,SIGMH,SIGML,SIG(5)
      COMMON/OPTION/KPRESM,QUAN(5),SELECT,IFIL(13)
      COMMON/PRO2/K(3),LENREF,LENUNC,K2(2),LOOPY,FIL(11),L4
      DATA CYCLE/5HCYCLE/
*               TLONG = NUMBER OF TARGETS USED
*               KANTEL = NUMBER OF TARGETS IN PHYSICAL BEAM OF ANTENNA
      NUMB = FLOAT(NUMB)/FLOAT(INCR) + .99
* NUMB IS NOW LENGTH OF REAMP AND REPHA
      ENTRY NTGET(NUMB,TLONG,KANTEL,*)                                      2
   10 NUMA = NUMB/KPRESM                                                    2
* NUMA IS NOW LENGTH OF REAMPS AND REAMPC
      LENUNC = FLOAT(KANTEL)/FLOAT(INCR) + .99                              3
*               LENUNC = LENGTH OF SEMI COMPRESSED BEAM (INCR ONLY)
      LENREF = LENUNC/KPRESM                                                4
*               LENREF = LENGTH OF FULLY COMPRESSED BEAM (INCR AND KPRESM)
      L4 = NUMA - LENREF                                                    5
*               L4 = NUMBER OF IMAGES TO BE GENERATED, LENGTH OF AMAP AND AMAPS
      NP = IFIX(FLOAT(KAN2*KANT)/FLOAT(INCR) + .99)/KPRESM                  6
*               NP = LENGTH OF EACH SUBAPERTURE (FULLY COMPRESSED)
*               NUMBER OF ARRAYS OF EACH LENGTH
      NUMB2 = MAX - (2*(LENREF + TLONG + NUMB) + MAX0(2*(NUMB + KANTEL),    7
     2      2*(NUMA + L4) + 3*NP))
*               NUMB2 = AMOUNT OF SURPLUS COMMON FOR THIS SET
*               MAX = AMOUNT OF COMMON AVAILABLE
      IF(NUMB2.LT.0) RETURN 1                                               8
*               CALCULATE POINTERS TO BE USED IN SETARG AND ELSEWHERE
* II IS POINTER TO SINANG, LENGTH LENREF
      II = 1                                                               11
* IJ IS POINTER TO COSANG, LENGTH LENREF
      IJ = II + LENREF                                                     12
* IA IS POINTER TO CELPHA, LENGTH TLONG
      IA = IJ + LENREF                                                     13
* IB IS POINTER TO CELAMP, LENGTH TLONG
      IB = IA + TLONG                                                      14
* ID IS POINTER TO REAMP, LENGTH NUMB
      ID = IB + TLONG                                                      15
* IE IS POINTER TO REPHA, LENGTH NUMB
```

```
          IE = ID + NUMB                                                       16
      * IF IS POINTER TO REAMPS, NOW LENGTH NUMB. LATER COMPRESSED TO LENGTH NUMA
          IF = IE + NUMB                                                       17
      * IG IS POINTER TO REAMPC, NOW LENGTH NUMB. LATER COMPRESSED TO LENGTH NUMA
          IG = IF + NUMB                                                       18
      * IH IS POINTER TO HGT1, LENGTH KANTEL
          IH = IG + NUMB                                                       19
      * IC IS POINTER TO ANTPHA, LENGTH KANTEL
          IC = IH + KANTEL                                                     20
      *             SET LOOPY = 1 IF STATISTICAL ANALYSIS OF TARGETS AND IMAGES
      *             IS TO BE PERFORMED
          LOOPY = 0                                                            21
          IF(SIGMH.EQ.SIGML .AND. SELECT.EQ.CYCLE) LOOPY = 1                   22
          RETURN                                                               25
          END                                                                  26
```

23727 WORDS OF MEMORY USED BY THIS COMPILATION

```
      SUBROUTINE ANTENA (CELPHA,CELAMP,REAMPS,REAMPC,ANTPHA,REAMP,
     2        REPHA,WGT,J7,J57,TLONG,IPAT,SINANG,COSANG)
*               THIS ROUTINE SIMULATES A RADAR ANTENNA
*               IT FIRST CALCULATES A PHASE ASSOCIATED WITH EACH TARGET
*               IN THE BEAM (ANTPHA). USING THIS IT CALCULATES THE SINE AND
*               COSINE REFERENCE FUNCTIONS TO BE USED IN IMAGING THE RETURNS, IT
*               THEN CALLS A SUBROUTINE WHICH ASSIGNS A WEIGHTING VALUE TO
*               EACH POSITION (TARGET) IN THE BEAM. USING THIS WEIGTHING
*               FCN, THE ANTENNA PHASE, THE TARGET AMPLITUDE, AND THE TARGET
*               PHASE IT CALCULATES A PAIR OF RETURNS (SINE AND COSINE)
*               FOR CERTAIN POSITIONS OF THE ANTENNA ACCORDING TO THE
*               SAMPLING RATE (INCR).
*               THEN USING THESE TWO RETURNS IT CALCULATES THE RETURN
*               AMPLITUDE (REAMP) AND RETURN PHASE (REPHA).
*               CELPHA = TARGET PHASE
*               CELAMP = TARGET AMPLITUDE
*               REAMPS = SINE CHANNEL OF RETURN
*               REAMPC = COSINE CHANNEL OF RETURN
*               ANTPHA = ANTENNA PHASE
*               REAMP = RETURN AMPLITUDE
*               REPHA = RETURN PHASE
*               WGT = PROCESSOR WEIGHTING FUNCTION
*               J7 = EQUIVALENT TO KANTEL
*               J57 = NUMBER OF RETURNS TO BE GENERATED
*               TLONG = NUMBER OF TARGETS
*               IPAT = SELECTS ANTENNA WEIGHTING FUNCTION TO BE USED
*               SINANG = SINE REFERENCE FUNCTION
*               COSANG = COSINE REFERENCE FUNCTION
      INTEGER TLONG
      DIMENSION CELAMP(TLONG),CELPHA(TLONG),REAMPS(J57),REAMPC(J57),
     2   ANTPHA(J7),REAMP(J57),REPHA(J57),WGT(J7),SINANG(1),COSANG(1)
      COMMON/OPTION/XPRESM,XP(8),FRAC3W,FIL(12)
*N3 = NUMBER OF TARGETS PER RESOLUTION CELL
*N1 = NUMBER OF RESOLUTION CELLS IN PHYSICAL BEAM
      COMMON/SIGCOM/ N1,KANTEL,N3,NP
      COMMON/PROCES/NAPS,KANT,INCR,LAP
**KANTEL = NUMBER OF TARGETS IN BEAM
**ANTEL = NUMBER OF RESOLUTION ELEMENTS IN BEAM
      ANTEL = N1
*CONST AND FAC5 = CONSTANTS USED TO CALCULATE ANTPHA
*BOTH ARE FUNCTIONS OF THE NUMBER OF ANTENNA RES CELLS AND TGTS/RES CELL
      CONST = 3.14159/ANTEL                                             2
      ANKAN = N3                                                        3
      FAC5 = -ANTEL/2.-1./(2.*ANKAN)                                    4
*LOOP CALCULATES ANTPHA OF EACH ELEMENT OF ANTENNA
*ANTPHA IS FUNCTION OF NUMBER OF ANTENNA ELEMENTS AND POSITION IN BEAM
      DO 66 M = 1,KANTEL                                                5
      AJ = M                                                            6
      ANTPHA(M) = CONST*(FAC5 + AJ/ANKAN)**2                            7
   66 CONTINUE                                                          8
*               THIS ENTRY,CALREF, IS USED BY THE TAPE I/O ROUTINE TO CALCULATE
```

```
*            ONLY THE SINE AND COSINE REFERENCE FCN, THE OTHER ARRAYS,
*            INCLUDING ANTPHA, HAVING ALREADY BEEN READ IN FROM TAPE.
      ENTRY CALREF(ANTPHA,SINANG,COSANG,J7)                           10
      ANKAN = N3                                                      10
      ANTEL = N1                                                      11
      AAA = (1.-ANTEL*ANKAN)/2.                                       12
      I2 = INCR*KPRESM                                                13
      I = 0                                                           14
      DO 55  M=1,KANTEL,I2                                            15
      R = M - 1                                                       16
*            FRACBW = DOPPLER MISMATCH REFERENCED TO RETURN SIGNAL BAND WIDTH
      TEMP = (AAA + R)/ANKAN*FRACBW*6.28318 + ANTPHA(M)               17
      I = I + 1                                                       18
*            SINE AND COSINE REFERENCE FCN TO BE USED IN PROCESSING OF RETURNS
      SINANG(I) = SIN(TEMP)                                           19
      COSANG(I) = COS(TEMP)                                           20
   55 CONTINUE                                                        21
*            THIS RETURN IS TAKEN IF ENTRY WAS MADE AT CALREF WITH J7 = 0
      IF(J7.EQ.0) RETURN                                              23
*            GENERATE ANTENNA WEIGHTING FUNCTION
      CALL WGTFCN (WGT,KANTEL,IPAT)                                   26
      KAN = 1                                                         27
      K3 = KANTEL + 1                                                 28
*            LOOP TO CALCULATE ONE PAIR OF RETURNS
      DO 99 J = 1,J57                                                 29
*            J3 = ONE TARGET PAST LEADING EDGE OF BEAM
      J3 = (J-1)*INCR + 2                                             30
      IF(J3.GT. TLONG + 1) KAN = J3 - TLONG                           31
*            IS LEADING EDGE PAST TARGETS. IF SO EFFECTIVELY CUT OFF FRONT END
*            OF BEAM THAT IS PAST TARGETS(IT CONTRIBUTES NOTHING)
  100 REAMPC(J) = 0.0                                                 34
      REAMPS(J) = 0.0                                                 35
*            LOOP SUMS ALL TARGETS AS SEEN BY BEAM IN CURRENT POSITION,
*            EACH TARGET IS WEIGHTED ACCORDING TO ITS POSITION IN THE BEAM
      DO 88  K = KAN,KANTEL                                           36
      K2A = J3 -K                                                     37
*            K2A RANGES OVER TARGETS SEEN BY BEAM
      K2B = K3 - K                                                    38
*            K2B RANGES OVER POSITIONS  (TARGETS) IN THE BEAM
      THETA = ANTPHA(K2B) + CELPHA(K2A)                               39
      REAMPC(J) = REAMPC(J) + CELAMP(K2A)*COS(THETA)*WGT(K2B)         40
      REAMPS(J) = REAMPS(J) + CELAMP(K2A)*SIN(THETA)*WGT(K2B)         41
      IF(K2A.EQ.1) GO TO 98                                           42
   88 CONTINUE                                                        45
*            USE TWO HALVES TO GET WHOLE RETURN
   98 REAMP(J) = SQRT(REAMPS(J)**2 + REAMPC(J)**2)                    47
      IF(REAMPC(J).EQ.0.) GO TO 97                                    48
*            FUNCTION  ATAN2  IS THE ARC TANGENT FUNCTION
      REPHA(J) = ATAN2(REAMPS(J),REAMPC(J))                           51
      GO TO 99                                                        52
*            THE SIGN FUNCTION IS DEFINED AS THE SIGN OF ARG2 TIMES THE
*            ABSOLUTE VALUE OF ARG1
```

```
      97  REPHA(J) = SIGN(1.57,REAMPS(J))                          53
      *              IF REAMPC IS ZERO REPHA BECOMES +1.57 OR -1.57 DEPENDING
      *              ON THE SIGN OF REAMPS
      99  CONTINUE                                                 54
          RETURN                                                   56
          END                                                      57

   23717 WORDS OF MEMORY USED BY THIS COMPILATION
```

```
      SUBROUTINE WGTFCN (WGT,N2,IPAT)
*              THIS ROUTINE GENERATES A WEIGHTING FUNCTION (WGT) OF LENGTH
*              N2 ACCORDING TO PATTERN SELECTED BY VALUE OF IPAT
*              WGT = ARRAY OF WEIGHTS TO BE GENERATED
*              IPAT = PATTERN SELECTOR FOR WEIGHT FUNCTION
      DIMENSION WGT(N2)
*              MAXPAT = NUMBER OF WEIGHTING FUNCTIONS AVAILABLE IN
*              WGTFCN (USED FOR ERROR CHECKING)
*              AT PRESENT IT IS TWO, IF USER ADDS HIS OWN FUNCTIONS HE MUST
*              CORRECT THE VALUE OF MAXPAT
      DATA MAXPAT/2/
      IF(IPAT.LE.MAXPAT) GO TO 100
      WRITE(6,99) IPAT                                                   4
   99 FORMAT (//18H *NON-FATAL ERROR*     //                             7
     2 36H WEIGHTING PATTERN SPECIFICATION OF    ,I3, 68H IS GREATER THAN
     3 NUMBER OF OPTIONS, CONSTANT WEIGHT OF 1. IS ASSUMED  ///)
      GO TO 1                                                            7
  100 GO TO (1,2,3),IPAT                                                 8
    1 DO 101 I=1,N2                                                      9
      WGT(I) = I.                                                       10
  101 CONTINUE                                                          11
      RETURN                                                            13
    2 SIZE = N2 - 1                                                     14
      HALF = SIZE/2. + 1.                                               15
      DO 102 I = 1,N2                                                   16
      AI = I                                                            17
      WGT(I) = .3162 + .683*COS(3.14159*(AI-HALF)/SIZE)                 18
  102 CONTINUE                                                          19
      RETURN                                                            21
    3 CONTINUE                                                          22
*              THE USER MAY WRITE HIS OWN WEIGHTING FUNCTION AND PLACE
*              IT HERE , HE MAY THEN ACCESS IT BY INPUTTING THE VALUE 3 FOR IPAT
*              OR KPAT (HE MUST ALSO CHANGE MAXPAT IN DATA STATEMENT ABOVE)
      RETURN                                                            23
*              IF THE USER WISHES TO ADD MORE THAN ONE FUNCTION HE MAY
*              DO SO BY MAKING THE FOLLOWING CHANGES
*              1) CHANGE VALUE OF MAXPAT IN DATA STANEMENT TO AGREE WITH
*              NUMBER OF FUNCTIONS NOW AVAILABLE
*              2) CHANGE STATEMENT LABELED 100 TO HAVE AS MANY ARGUMENTS AS
*              THERE ARE FUNCTIONS, THE ARGUMENTS ADDED WILL BE THE STATEMENT
*              NUMBERS OF WHERE EACH FUNCTION BEGINS
*              3) WRITE THE FUNCTION DESIRED, THE FIRST STATEMENT OF WHICH
*              MUST CONTAIN A LABEL WHICH APPEARS IN THE COMPUTED GO TO
*              OF STATEMENT 100 AND THE LAST STATEMENT MUST BE A RETURN
*              TO ACCESS THE NEW WEIGHTING FUNCTION INPUT AS THE WEIGHTING
*              NUMBER THE POSITION OF THE LABEL OF THE FUNCTION IN THE COMPUTED
*              GO TO STATEMENT 100
*
*              IF USER WISHES TO INPUT WEIGHTS VIA CARDS ON THE SYSTEM
*              INPUT FILE, 05, THESE CARDS SHOULD COME AFTER THE LAST
*              REQUIRED PROGRAM DATA CARD AND AFTER ANY CARDS THAT WILL BE
```

```
 52   *              INPUT BY SUBROUTINE EXTARG, IF USER INPUTS TARGETS BY CARDS.
 53   *              IF CARDS ARE TO BE READ FOR BOTH THE ANTENNA AND REFERENCE
 54   *              FUNCTION WEIGHTING THE ORDER SHOULD BE -- EXTARG TARGETS,
 55   *              ANTENNA WEIGHTS REFERENCE FUNCTION WEIGHTS
 56        END                                                                   24

   23509 WORDS OF MEMORY USED BY THIS COMPILATION
```

```
      SUBROUTINE COROMP
*              ROUTINE TO OUTPUT ARRAYS USED IN CURRENT DATA SET
*              SUMMARY OF DATA SET OUTPUT AT ENTRY POINT OUT2
*              ONE ARRAY AT A TIME MAY BE OUTPUT AT ENTRY POINT SNAP
      INTEGER TLONG
      DIMENSION LEN(15),ANAME(45),HEAD(3),ARRAY(1),T(4)
      COMMON CELL(1)
      COMMON/CELENG/TLONG,NUMA,NUMB,INUMB(26)
      COMMON/SIGCOM/KAN1,KANTEL,KAN2,NP
      COMMON/RECEIV/ GAIN,SAT,ALGAIN,STN
      COMMON/PROCES/ NAPS,KANT,INCR,LAP
      COMMON/OPTION/ KPRESM,IQUANT,RANG1,RANG2,PLOT,PUNCH,SEL,FRACBW,
     2  NUMPLT,KQUANT,KPAT,CORE,NCHAN,PUNTAR,RCA,AVEAMP,FIL(4)
      COMMON/PRO2/KFIRST,KPROC,KSAMPD,LENREF,LENUNC,IFIL(14),L4
      COMMON/APERTJ/AWGTS(100)
      DATA YES,NALL/3HYES,3HALL/
      DATA (T(I),I=1,4) /6HTAKEN ,6H      ,6HNOT TA ,6HKEN   /
*              MUSE = NUMBER OF ARRAYS TO BE OUTPUT
      DATA MUSE/13/
*              TITLES OF ARRAYS OUTPUT
      DATA (ANAME(I),I=1,39,3)/
     1 18HSINE REF. FCN.      ,18HCOSINE REF. FCN.    ,
     2 18HTARGET PHASE        ,18HTARGET AMPLITUDE    ,
     3 18HRETURN AMPLITUDE    , 18HRETURN PHASE       ,
     4 18HRETURN AMP(SINE)    , 18HRETURN AMP(COSINE) ,
     5 18HPROCESSOR WGT FCN   , 18HINTR WGT REF FCN/S ,
     6 18HINTR WGT REF FCN/C  ,18HFINAL IMAGE         ,
     7 18HIMAGE FOR LAST AP   /
*              ABOVE DATA STATEMENT DEFINES TITLES TO BE OUTPUT AT TOP OF EACH
*              ARRAY
*              SET UP ARRAY OF LENGTHS OF EACH ARRAY
      LEN(1) = LENREF
      LEN(2) = LENREF
      LEN(3) = TLONG
      LEN(4) = TLONG
      LEN(5) = NUMB
      LEN(6) = NUMB
      LEN(7) = NUMA
      LEN(8) = NUMA
      LEN(9) = NP
      LEN(10) = NP
      LEN(11) = NP
      LEN(12) = L4
      LEN(13) = L4
      WRITE(6,10)
   10 FORMAT(27H1CORE DUMP OF ALL CORE USED    //)
      U22 = 0
      DO 99 I = 1,MUSE
      J2 = 3*I
      J1 = J2 - 2
      WRITE(6,20) (ANAME(K),K=J1,J2),LEN(I)
```

```
      20 FORMAT(//1X,3A5,10X,9HLENGTH = ,I5//)                           26
         J11 = J22 + 1                                                    26
         J22 = J11 + LEN(I) - 1                                           27
         WRITE(6,30) (CELL(K),K=J11,J22)                                  28
      30 FORMAT (1X,10F12.3)                                              33
      99 CONTINUE                                                         33
         RETURN                                                           35
         ENTRY SNAP (ARRAY,HEAD,LONG)                                     36
 *              ENTRY POINT TO OUTPUT ONE ARRAY WITH HEADING
 *              ARRAY = ARRAY TO BE OUTPUT
 *              HEAD = THREE WORD LITERAL FOR HEADING (MUST BE EXACTLY 3 WORDS)
 *              LONG = LENGTH OF ARRAY
         WRITE(6,20) (HEAD(K),K=1,3),LONG                                 36
         WRITE(6,30) (ARRAY(K),K=1,LONG)                                  42
         RETURN                                                           47
         ENTRY OUT2(IPAT)                                                 48
 *              ENTRY POINT OUT2 OUTPUTS A SUMMARY OF THE CURRENT DATA SET
         WRITE(6,110) KAN1,KAN2,KANTEL,FRACBW,IPAT                        48
     110 FORMAT (//22H ANTENNA SPECIFICATION    /                         51
        2   49H NUMBER OF RESOLUTION CELLS IN BEAM OF ANTENNA =    ,I5/
        3  41H NUMBER OF TARGETS PER RESOLUTION CELL =    ,I3/
        4  47H TOTAL NUMBER OF TARGETS IN THE ANTENNA BEAM =    ,I5/
        5 59H DOPPLER MISMATCH REFERENCED TO RETURN SIGNAL BAND WIDTH =  ,
        6  F5.2/ 21H WEIGHTING FUNCTION # ,I3,24H WAS USED ON THE ANTENNA)
         WRITE(6,120) GAIN,SAT,ALGAIN,STN,RCA                             51
     120 FORMAT (//23H RECEIVER SPECIFICATION    /                        54
        2   38H GAIN OF RECEIVER OVER LINEAR RANGE =    ,F6.2/
        3 37H MAXIMUM MAGNITUDE OF LINEAR RANGE =     ,F7.2/
        4 42H GAIN OF RECEIVER OVER LOGRITHMIC RANGE =   ,F6.2/
        5 25H SIGNAL TO NOISE RATIO =    ,F6.2/
        6 49H MEAN VOLTAGE REFLECTION COEFFICIENT AMPLITUDE =   ,F6.2)
         WRITE(6,130) KPROC,NCHAN,NAPS,KFIRST,KANT,LAP,INCR,NP,KPAT      54
     130 FORMAT(// 24H PROCESSOR SPECIFICATION    /                       57
        2 22H TYPE OF PROCESSING =   ,A4,8H FOCUSED    /
        2 41H NUMBER OF CHANNELS USED IN PROCESSING =   ,I1/
        2 48H NUMBER OF APERTURES TO BE USED IN PROCESSING =   ,I3/
        3 63H THE FIRST APERTURE STARTS WITH FULLY COMPRESSED RETURN NUMBER
        3    ,I4/
        3 44H SIZE OF EACH APERTURE (RESOLUTION CELLS) =     ,I5/
        4 57H OVERLAP BETWEEN ADJACENT APERTURES (RESOLUTION CELLS) =   ,I3/
        5 63H NUMBER OF TARGETS INCREMENTED PER RETURN (ANALOGOUS TO PRF) =
        5     ,I3/
        6 38H NUMBER OF RETURNS IN EACH APERTURE =      ,I5/
        7 21H WEIGHTING FUNCTION # ,I3,52H WAS USED ON THE REFERENCE FUNCTI
        8ON OF EACH APERTURE    )
         WRITE(6,145) (AWGTS(I),I=1,NAPS)                                 57
     145 FORMAT(//48H WEIGHTS FOR EACH SUBAPERTURE ARE AS FOLLOWS --   //  62
        2     (1X,10F10.2))
         IF(KPRESM.LE.1) GO TO 141                                        62
         WRITE(6,140) KPRESM                                              65
     140 FORMAT(//28H THE PRESUM OPTION WAS TAKEN/                        68
        2 1X,I3,38H RETURNS WERE SUMMED BEFORE PROCESSING)
```

```
              GO TO 161                                                      68
        141   WRITE (6,150)                                                  69
        150   FORMAT (// 32H THE PRESUM OPTION WAS NOT TAKEN)                71
        161   IF(IQUANT.EQ.36) GO TO 171                                     71
              IQ = 2**IQUANT                                                 74
              WRITE(6,170) IQUANT,IQ,RANG1,RANG2                             75
        170   FORMAT(// 43H THE QUANTIZE OPTION FOR RETURNS WAS TAKEN    /   78
             2 53H NUMBER OF BITS AVAILABLE TO STORE QUANTIZED VALUE =  ,I3/
             3 32H NUMBER OF QUANTIZATION LEVELS =  ,I10/
             4 37H LOWER LIMIT OF QUANTIZATION RANGE =  ,F7.2/
             4 37H UPPER LIMIT OF QUANTIZATION RANGE =  ,F7.2)
              GO TO 175                                                      78
        171   WRITE(6,180)                                                   79
        180   FORMAT(// 50H THE QUANTIZATION OPTION FOR RETURNS WAS NOT TAKEN  )  81
        175   IF(KQUANT.EQ.35) GO TO 178                                     81
              IQ = 2**KQUANT                                                 84
              WRITE(6,177) KQUANT,IQ                                         85
        177   FORMAT(61H THE QUANTIZATION OPTION FOR THE REFERENCE FUNCTION WAS  88
             2TAKEN   /
             3 53H NUMBER OF BITS AVAILABLE TO STORE QUANTIZED VALUE =  ,I3/
             4 32H NUMBER OF QUANTIZATION LEVELS =  ,  I10)
              GO TO 182                                                      88
        178   WRITE (6,179)                                                  89
        179   FORMAT(66H THE QUANTIZATION OPTION FOR THE REFERENCE FUNCTION WAS  91
             2NOT TAKEN     )
        182   IF(PLOT.NE.YES) GO TO 172                                      91
              N27 = I                                                        94
              IF(NAPS.EQ.1 .OR. NUMPLT.NE.NALL) GO TO 189                    95
              N27 = NAPS + 1                                                 98
        189   WRITE(6,190) N27                                               99
        190   FORMAT(//27H THE PLOT OPTION WAS TAKEN,  I5,39H PAGE(S) WERE PLOTT  102
             2ED FOR THIS DATA SET      )
              GO TO 191                                                      102
        172   WRITE(6,200)                                                   103
        200   FORMAT(//30H THE PLOT OPTION WAS NOT TAKEN)                    105
        191   K = 3                                                          105
              N = 3                                                          106
              IF(PUNCH.EQ.YES) N = 1                                         107
              IF(PUNTAR.EQ.YES) K = 1                                        110
              WRITE(6,220) T(N),T(N+1),T(K),T(K+1)                           113
        220   FORMAT(//28H THE IMAGE PUNCH OPTION WAS  ,2A6/                 116
             2  29H THE TARGET PUNCH OPTION WAS  ,2A6)
              K = 3                                                          116
              IF(CORE.EQ.YES) K = 1                                          117
              WRITE(6,230) T(K),T(K+1)                                       120
        230   FORMAT(//25H THE CORE DUMP OPTION WAS  ,2A6)                   123
        211   RETURN                                                         123
              END                                                            124

     23678 WORDS OF MEMORY USED BY THIS COMPILATION
```

```
      SUBROUTINE NOISY(REAMP,REPHA,REAMPS,REAMPC,ANTPHA,WGT1,NUMB,
     2                 KANTEL,CELPHA)
*                FUNCTION OF THIS ROUTINE IS TO ADD NOISE TO THE RETURN
*                AMPLITUDE AND APPLY LINEAR AND LOGRITHMIC GAIN, IF ANY.
*                NOISE IS ALWAYS ADDED TO THE REAMP UNLESS THE SIGNAL TO NOISE
*                RATIO (STN) = 999.99
*                IF NO LINEAR OR LOGRITHMIC GAIN IS TO BE ADDED  GAIN = 1.
*                SAT = 999.  AND ALGAIN = 0.
*                ALL NOISE AND GAIN IS ADDED TO THE ONE SIGNAL, RETURN AMPLITUDE,
*                THIS SIGNAL IS THEN SPLIT IN TWO FOT FORM THE SINE AND
*                COSINE HALVES OF THE SIGNAL
      DIMENSION REAMP(NUMB),REPHA(NUMB),REAMPS(NUMB),REAMPC(NUMB)
     2          ,ANTPHA(KANTEL),WGT1(KANTEL),CELPHA(1)
      COMMON/RECEIV/ GAIN,SAT,ALGAIN,STN
      COMMON/OPTION/A(6),SELECT,B(7),RCA,AVEAMP,C2(4)
      COMMON/PXCH/C(2),SIGMA,SHORT,D,CONAMR,CONPHA,E(2)
      COMMON/PRO2/AY(4),LENUNC,A3,KSAMP2,LOOPY,FIL(12)
*                GAIN = LINEAR GAIN OF RECEIVER
*                SAT = SATURATION OR CLIPPING POINT
*                ALGAIN = LOGRITHMIC GAIN OF RECEIVER
*                STN = SIGNAL TO NOISE FIGURE FOR THE RECEIVER
*                RNOISE = MULTIPLIER FOR RANDOM NOISE
*                AMP = AVERAGE RETURN
      DATA EXTARG,CONSTA,CPHASE/6HEXTARG,6HCONSTA,6HCPHASE/
      DATA NST/8516878731/
      RNOISE = 0.
      IF(STN.EQ.999.99) GO TO 50                                          2
*                THIS SECTION CALCULATES THE MULTIPLIER  RNOISE  ACCORDING
*                TO THE SIGNAL TO NOISE RATIO GIVEN AND THE DISTRIBUTION OF
*                THE TARGET'S AMPLITUDE AND PHASE
      IF(SELECT.NE.EXTARG) GO TO 100                                      5
      AMP = AVEAMP                                                        8
      IF(AVEAMP.EQ.0.) WRITE(6,101)                                       9
  101 FORMAT(//18H *NON-FATAL ERROR*    /                                13
     2 58H USER MUST SUPPLY AVERAGE RETURN AMPLITUDE, NO NOISE ADDED    )
      GO TO 15                                                           13
  100 IPP = 1                                                            14
      IF(CONPHA.EQ.CPHASE) IPP = 3                                       15
      IAP = 0                                                            18
      IF(CONAMP.EQ.CONSTA)IAP = 1                                        19
      IAPP = IPP + IAP                                                   22
*                IAPP = 1       STAT. AMPLITUDE AND STAT. PHASE
*                IAPP = 2       STAT. AMPLITUDE AND CONSTANT PHASE
*                IAPP = 3       CONSTANT AMPLITUDE AND STAT. PHASE
*                IAPP = 4       CONSTANT AMPLITUDE AND CONSTANT PHASE
      GO TO (1,1,3,4),IAPP                                               23
    1 DO 199 I=1,KANTEL                                                  24
      AMP = AMP + WGT1(I)**2                                             25
  199 CONTINUE                                                           26
      IF(IAPP.EQ.1) AMP = 2.*AMP                                         28
      GO TO 20                                                           31
```

```
    3 DO 88 I=1,KANTEL
      AMP = AMP + WGT1(I)**2
      DO 77 J=I,KANTEL
      IF(I.EQ.J) GO TO 77
      TEMP = TEMP + WGT1(I)*WGT1(J)*COS(ANTPHA(I)-ANTPHA(J))
   77 CONTINUE
   88 CONTINUE
      AMP = 2.*AMP + 1.25**2*TEMP
      GO TO 20
    4 A1 = CELPHA(1)
      DO 66 I=1,KANTEL
      THETA = ANTPHA(I) + A1
      TEMP = TEMP + WGT1(I)*COS(THETA)
      AMP = AMP + WGT1(I)*SIN(THETA)
   66 CONTINUE
      AMP = TEMP**2 + AMP**2
   20 AMP = AMP*RCA**2
*               RNOISE = MULTIPLIER FOR ADDITIVE SYSTEM NOISE
   15 RNOISE = SQRT(AMP/(2.*STN))
*               CLIP = MAXIMUM OUTPUT OF LINEAR RANGE
   50 IF(GAIN.EQ.1. .AND. SAT.EQ.999. .AND. RNOISE.EQ.0. .AND. ALGAIN
     2   .EQ.0.) GO TO 95
      IF(RNOISE.EQ.0.) GO TO 51
      DO 186  I=1,NUMB
      A1 = RNOISE*RMS(NST)
      A2 = RNOISE*RMS(NST)
      TEMP = REAMP(I) + A1
      REPHA(I) = REPHA(I) + ATAN2(A2,TEMP)
      REAMP(I) = SQRT(TEMP**2 + A2**2)
  186 CONTINUE
   51 CLIP = GAIN*SAT
      SAT2 = 1.- SAT
      DO 99 I=1,NUMB
      ATEMP = REAMP(I)
*               IF RETURN IS LESS THAN OR EQUAL TO CLIPPING POINT COMPUTE
*               SIMPLE LINEAR GAIN, IF NOT COMPUTE LINEAR GAIN PLUS LOGRITHMIC
*               GAIN
      IF(ATEMP.LE.SAT) GO TO 5
      ATEMP = CLIP + ALGAIN*ALOG10(SAT2 + ATEMP)
      GO TO 10
    5 ATEMP = GAIN*ATEMP
*               SPLIT RETURN INTO TWO CHANNELS SINE AND COSINE
   10 REAMPS(I) = SIN(REPHA(I))*ATEMP
      REAMPC(I) = COS(REPHA(I))*ATEMP
   99 CONTINUE
   95 IF(LOOPY.NE.1) RETURN
      WRITE(6,11)
   11 FORMAT(/' NOISE HAS BEEN ADDED TO THE RETURNS IF SPECIFIED'  )
      CALL HENVAR(KSAMP2,LENUNC+1,REAMP,'RETURNS',NUMB-LENUNC)
      RETURN
      END
```

```
      SUBROUTINE PRESUM(REAMPS,I)
C***THIS SUBROUTINE SUMS THE DESIRED NUMBER OF RETURNS, COMPRESSING THE ARRAYS AS
C*THEY ARE SUMMED.  IT ALSO RECALCULATES THE POINTERS FOR THE NEXT TWO ARRAYS
C*BASED ON THE LENGTH OF THE NOW COMPRESSED ARRAYS
C*               METHOD--THE FIRST  I  ELEMENTS OF THE FIRST ARRAY ARE
C*               SUMMED AND THE VALUE STORED IN ELEMENT ONE OF THE ARRAY
C*               THEN THE NEXT I ELEMENTS ARE SUMMED AND THE VALUE STORED
C*               IN THE SECOND ELEMENT OF THE ARRAY AND SO ON FOR THE FIRST
C*               ARRAY, ITS SIZE BEING REDUCED BY A FACTOR OF I.  IF ON THE
C*               LAST ROUND FOR THAT ARRAY I ELEMENTS ARE NOT AVAILABLE
C*               TO BE SUMMED THEY ARE DISCARDED.  THEN THE FIRST  I
C*               ELEMENTS OF THE SECOND ARRAY ARE SUMMED AND ITS VALUE
C*               STORED IN THE LOCATION IMMEDIATELY FOLLOWING THE LAST
C*               LOCATION USED BY THE NOW COMPRESSED FIRST ARRAY AND SO ON
C*               UNTIL THE SECOND ARRAY IS COMPLETED
C*               REAMPS = ARRAY TO BE PRESUMMED
C*               I = NUMBER OF RETURNS TO BE PRESUMMED
      DIMENSION REAMPS(1)
      COMMON/CELENG/N5(2),NUMB,IAFIL(5),IF,IG,IH,INUMB(18)
      I1 = I - 1
      N = NUMB + 1 - I
      NB = 0
      ICON = 0
    5 DO 99 J=1,N,I
      J2 = NB + J
      ICON = ICON + 1
      DO 88 K = 1,I1
      M = J2 + K
      REAMPS(ICON) = REAMPS(J2) + REAMPS(M)
   88 CONTINUE
   99 CONTINUE
      IF(NB.NE.0) RETURN
      NB = NUMB
C*               RECALCULATE POINTERS
C* IG IS NOW POINTER TO COMPRESSED ARRAY REAMPC
      IG = IF + ICON
C* IH IS NOW POINTER TO WST2
      IH = IG + ICON
      GO TO 5
      END
```

23706 WORDS OF MEMORY USED BY THIS COMPILATION

```
      SUBROUTINE QUANT (REAMPS,NTWICE)
*              FUNCTION OF THIS SUBROUTINE IS TO QUANTIZE THE RETURNS
*              BEFORE PROCESSING.  QUANT ASSIGNS TO EACH ELEMENT IN EACH CHANNEL
*              OF THE RETURN AN INTEGER NUMBER (REAL TYPE) CORRESPONDING TO
*              THE QUANTUM IN WHICH THE REAMP FALLS, ANY VALUE OF REAMP FALLING
*              OUTSIDE THE RANGE OF THE QUANTIZATION WILL BE ASSIGNED 0 IF LOW
*              OR ALEVEL-1 IF HIGH,  WHERE ALEVEL IS THE NUMBER OF QUANTIZATION
*              LEVELS
*              QUANT TREATS REAMPS AND REAMPC AS ONE LONG ARRAY TO BE QUANTIZED
*              REAMPS = ARRAY TO BE QUANTIZED
*              NTWICE = LENGTH OF ARRAY TO BE QUANTIZED, IN THIS CASE IT IS THE
*              COMBINED LENGTH OF REAMPS AND REAMPC
*              IQUANT = NUMBER OF BITS AVAILABLE TO STORE QUANTIZED VALUE
*              RANG1 = LOWER MAGNITUDE OF RANGE TO BE QUANTIZED
*              RANG2 = UPPER MAGNITUDE OF RANGE TO BE QUANTIZED
      DIMENSION REAMPS(NTWICE)
      COMMON/OPTION/KPRESM,IQUANT,RANG1,RANG2,FIL(16)
      ALEVEL = 2.**(IQUANT-1)
*ALEVEL IS THE NUMBER OF QUANTIZATION LEVELS ON EACH SIDE OF ZERO
      QUANSI = (RANG2 - RANG1)/(ALEVEL-1.)                                  2
*QUANSI IS THE SIZE OF EACH QUANTUM
    2 DO 99 I = 1,NTWICE                                                    3
      RESIGN = SIGN(1.,REAMPS(I))                                           4
      AMAG = ABS(REAMPS(I))                                                 5
      IF(AMAG.LE.RANG1) GO TO 5                                             6
      IF(AMAG.GE.RANG2) GO TO 15                                            9
      AMAG = AINT((AMAG - RANG1)/QUANSI)                                   12
      GO TO 98                                                             13
    5 AMAG = 0.                                                            14
      GO TO 98                                                             15
   15 AMAG = ALEVEL - 1.                                                   16
*              PUT SIGN BACK ON
   98 REAMPS(I) = RESIGN*AMAG                                              17
   99 CONTINUE                                                             18
      RETURN                                                               20
      END                                                                  21
```

23643 WORDS OF MEMORY USED BY THIS COMPILATION

```
      SUBROUTINE DIGPRO(SINANG,COSANG,CELPHA,CELAMP,REAMP,REPHA,REAMPS,
     2  REAMPC,AMAP,AMAPS,SPROD,CPROD,WGT2,L1,L3,L4,KTGS,L2,NUMB)
      DIMENSION CELPHA(KTGS),CELAMP(KTGS),REAMPS(L1),REAMPC(L1),
     2          AMAP(L1),AMAPS(L1),SINANG(1),COSANG(1)     ,
     3            SPROD(L3),CPROD(L3),WGT2(L3),REAMP(NUMB),REPHA(NUMB)
      COMMON/PROCES/ NAPS,KSBRSL,KPULSD,LAP
      COMMON/SIGCOM/KBRESL,KBTGS,KTPRES,K57
      COMMON/PRO2/KFIRST,KPROC,KSAMPD,LENREF,LENUNC,KSAMP1,KSAMP2,
     2   LOOPY,IFIL(12)
      COMMON/OPTION/KPRESM,QUAN(3),PLOT,PUNCH,SELECT,FRACBW,NUMPLT,
     2  KQUANT,KPAT,DUMP,NCHAN,PUNTAR,FIL(6)
      COMMON/APERTJ/APWGTS(100)
      DATA NALL/3HALL/
      DATA YES,CYCLE/3HYES,5HCYCLE/
      DATA IS2/4HZONE/
*               L1 = TOTAL NUMBER OF RETURNS  (IMAGE + REAL AP - 1)
*               L2 = NUMBER OF RETURNS IN REAL BEAM
*               L3 = NUMBER OF RETURNS PER SUBAPERTURE
*               L4 = NUMBER OF IMAGE RETURNS
*               APWGTS = APERTURE WEIGHTS
*               KBRESL = NUMBER OF RES CELLS IN THE REAL BEAM FULL FOCUSED
*               KTPRES = NUMBER OF TARGETS PER RES CELL
*               KSBRSL = NUMBER OF RESOLUTION CELLS IN SUBAPERTURE LENGTH INTEGER
*               KPULSD = NUMBER OF TARGETS WHICH THE REAL BEAM MOVES PER PULSE
*               KFIRST = IMAGE ELEMENT IN THE REAL BEAM ON WHICH THE FIRST SUBAP
*               STARTS, IT TELLS HOW MUCH THE FIRST SUBAP IS SQUINTED BACK
*               LAP = NUMBER OF FULLY COMPRESSED RETURNS LAPPED IN EACH SUBAP
*               NAPS = NUMBER OF APERTURES OR RATHER SUBAPERTURES
*               KSAMPD = INCEMENT OF IMAGE VALUES TO FIND IMAGE S/N
*               KPROC CAUSES PROCESSING TO BE EITHER 'FULL' FOCUS OR 'SEMI'FOCUS
*               KBTGS = NUMBER OF TARGETS IN REAL BEAM
*               KSBTGS = NUMBER OF TARGETS IN SUBAPERTURE
*               KQUANT = NUMBER OF BITS AVAILABLE TO STORE WEIGHTED REF FCN
      KSBTGS = KTPRES*KSBRSL
  300 LB = 0                                                            DIG   2
      I2 = KPRESM*KPULSD                                                DIG   3
      ATEMP = 0.                                                              4
*               GENERATE PROCESSOR WEIGHTING FUNCTION FOR FULL LENGTH OF
*               SUBAPERTURE USED
      CALL WGTFCN(WGT2,KSBTGS,KPAT)                                           5
*               COMPRESS REF FUNCTION PHASE ARRAY TO AGREE W/ COMP, RETURNS
      ATEMP = WGT2(1)                                                         6
      DO 1 I=1,KSBTGS,I2                                                DIG   7
      LB = LB + 1                                                       DIG   8
      IF(ATEMP.LT.WGT2(I)) ATEMP = WGT2(I)                                    9
    1 WGT2(LB) = WGT2(I)                                                DIG  12
      ATEMP * 1./ATEMP                                                       14
*               NORMALIZE WEIGHTING FUNCTION TO INTERVAL  0.-1.
      DO 3 I = 1,LB                                                          15
    3 WGT2(I) = WGT2(I)*ATEMP                                                16
*               IF FULL FOCUS SKIP TO 15
```

```
                IF(KPROC.NE.IS2) GO TO 15                                    18
      *                   FOR SEMI FOCUS PROCESSING USE ONLY SIGN OF REFERENCE FUNCTION
                DO 16 I=1,L2                                                 21
                IF(SINANG(I).NE.0.) SINANG(I) = SIGN(1.,SINANG(I))           22
             16 IF(COSANG(I).NE.0.) COSANG(I) = SIGN(1.,COSANG(I))           25
             15 DO 22 I=1,L4                                                 29
                AMAPS(I) = 0.                                     6/5/70A    30
             22 CONTINUE                                          6/5/70A    31
                L9 = L3 + 1                                                  33
                ALEVEL = 2.**(KQUANT-1)                                      34
      *                   LOOP IMAGES RETURNS FOR ONE APERTURE
             73 DO 72 J=1,NAPS                                               35
                N = LENREF - J*L3 + (J-1)*LAP - KFIRST + 1                   36
                NFIRST = J*L3 - (J-1)*LAP + KFIRST                           37
      *                   N = ONE LESS THAN NUMBER OF FIRST RETURN TO BE PUT IN PROCESSOR
      *                   NFIRST = ONE GREATER THAN NUMBER OF LAST REF FUNCTION TO BE
      *                   USED FOR THIS APERTURE
                IF(N.GE.0) GO TO 23                                          38
      *                   IF N IS NEGATIVE CURRENT APERTURE DOES NOT EXIST AND THEREFORE
      *                   IS IN ERROR
                IF(NAPS.EQ.1) GO TO 75                                       41
                WRITE(6,42) J                                                44
             42 FORMAT(18H *NON-FATAL ERROR*    //                           47
               2  11H APERTURE #    ,I3,57H AND ALL SUCCEEDING APERTURES ARE ILLEGA
               3L AND ARE DELETED    //)
                NAP = J - 1                                                  47
                GO TO 74                                                     48
             75 WRITE(6,43)                                                  49
             43 FORMAT (14H *FATAL ERROR*  //                                51
               2 63H ALL APERTURES ARE ILLEGAL, IMAGING OF THIS DATA SET IS DELETE
               3D    )
                RETURN                                                       51
      *                   CALCULATE WEIGHTED REFERENCE FUNCTION FOR CURRENT APERTURE
             23 DO 47 I=1,L3                                                 52
                L = NFIRST - I                                               53
                LL = L9 - I                                                  54
                SPROD(I) = SINANG(L)*WGT2(LL)                                55
                CPROD(I) = COSANG(L)*WGT2(LL)                                56
             47 CONTINUE                                                     57
                IF(KQUANT.EQ.36) GO TO 48                                    59
      *                   QUANTIZE REFERENCE FUNCTION* IF REQUIRED
                DO 49 I = 1,L3                                               62
                SPROD(I) = IFIX(SPROD(I)*ALEVEL)                             63
                CPROD(I) = IFIX(CPROD(I)*ALEVEL)                             64
             49 CONTINUE                                                     65
      *                   LOOP IMAGES RETURN FOR CURRENT APERTURE
             48 DO 71 M = 1,L4                                               67
                CC=0.                                                        68
                CS=0.                                                        69
                SC=0.                                                        70
                SS=0.                                                        71
                DO 61 I=1,L3                                          DIG    72
```

```
              LLL = N + I                                                        73
              CC = CC + REAMPC(LLL)*CPROD(I)                                     74
      *                IF 1 CHANNEL PROCESSING COMPUTE ONLY ONE OF FOUR
              IF(NCHAN.EQ.1) GO TO 61                                            75
              CS = CS + REAMPC(LLL)*SPROD(I)                                     78
              SC = SC + REAMPS(LLL)*CPROD(I)                                     79
              SS = SS + REAMPS(LLL)*SPROD(I)                                     80
         61   CONTINUE                                                           81
              N = N + 1                                                          83
              AMAPAD=CC+SS                                                       84
              AMAPSU=SC-CS                                                       85
              AMAP(M) = SQRT(AMAPAD**2 + AMAPSU**2)                              86
      *                PERFORM WEIGHTED SUM FOR FINAL RESULT
         71   AMAPS(M) = AMAPS(M) + AMAP(M)*APWGTS(J)                            87
              WRITE(6,1001) J,(AMAP(I),I=1,L4)                                   89
       1001   FORMAT(24H1RESULTS FROM APERTURE #  I3/(1X,10F10.3))               95
      *                PERFORM STATISTICAL ANALYSIS, IF REQUIRED
              IF(LCOPY.EQ.1) CALL MENVAR(KSAMPD,1,AMAP,'CURRENT AP',L4)          95
        404   CONTINUE                                                           98
              IF(PLOT.EQ.YES .AND. (NUMPLT.EQ.NALL .OR. J.EQ.1)) CALL OURPLT     99
             2 (CELPHA,CELAMP,REPHA,REAMP,AMAP,KTGS,L4,NUMB,LENUNC,J)
         72   CONTINUE                                                           102
              NAP = NAPS                                                         104
         74   IF(NAP .EQ. 1) GO TO 400                                           105
              WRITE(6,280) NAP,(AMAPS(I),I=1,L4)                                 108
        280   FORMAT(22H1SUMMATION OF PREVIOUS  ,I5,28H APERTURES FOR FINAL RESU 114
             2LTS      /(1X,10F10.3))
              IF(PLOT.EQ.YES) CALL OURPLT (CELPHA,CELAMP,REPHA,REAMP,AMAPS,KTGS, 114
             2 L4,NUMB,LENUNC,0)
              IF(LCOPY.EQ.1) CALL MENVAR(KSAMPD,1,AMAPS,'FINAL IMAGE',L4)        117
        400   LOOPY = 0                                                          120
      *                PUNCH FINAL IMAGE IF REQUIRED
              IF(PUNCH.EQ.YES) WRITE (43,1003) L4,(AMAPS(I),I=1,L4)              121
       1003   FORMAT(I10,7F10.3/(8F10.3))                                        129
              RETURN                                                     6/5/70A 129
              END                                                                130

    23775 WORDS OF MEMORY USED BY THIS COMPILATION
```

```
      SUBROUTINE OURPLT (CELPHA,CELAMP,REPHA,REAMP,AMAP,KTAREL,L4,
     2         NUMB,LENUNC,J)
*              THIS ROUTINE GRAPHS FIVE ARRAYS, ONE ABOVE THE OTHER IN
*              THE FORM OF AN 8.5 BY 11. INCH PAGE.  THE ARRAYS ARE IN THIS
*              ORDER FROM THE TOP--
*              1) CELPHA - TARGET PHASE
*              2) CELAMP - TARGET AMPLITUDS
*              3) REPHA  - RETURN PHASE
*              4) REAMP  - RETURN AMPLITUDE
*              5) AMAP   - IMAGE AFTER PROCESSING
*
*              THE ROUTINES PLOT AND INPLOT ARE USED FOR THE PLOTTING
      DIMENSION CELPHA(KTAREL),CELAMP(KTAREL),REAMP(NUMB),REPHA(NUMB),
     2    AMAP(L4)
      YYYYY = KTAREL + 2
      XXXXX = LENUNC - 2                                                2
      PPP = NUMB + 2                                                    3
      XXX = -1                                                          4
      LENREF = 1                                                        5
      PPPPP = L4 + 2                                                    6
      BTEL = LENREF                                                     7
      ANTEL = LENUNC                                                    8
      VIMAG = 0.                                                        9
*              FIND MAXIMUM OF ARRAYS, MINIMUMS ARE ZERO
*              IF NOT FIRST ENTRY FOR THIS SET ALREADY HAVE MAXIMUM FOR
*              TARGETS AND RETURNS
      IF(J.NE.1) GO TO 81                                               10
      TARG=0.                                                           13
      VRET=0.                                                           14
*              FIND MAXIMUM OF ARRAYS, MINIMUMS ARE ZERO
      DO  84  I=1,KTAREL                                                15
      IF (CELAMP(I) .GT. TARG) TARG=CELAMP(I)                           16
   84 CONTINUE                                                          19
      DO 82 I = LENUNC,NUMB                                             21
      IF(REAMP(I).GT.VRET) VRET=REAMP(I)                                22
   82 CONTINUE                                                          25
   81 DO 83 I=LENREF,L4                                                 27
      IF(AMAP(I) .GT. VIMAG) VIMAG = AMAP(I)                            28
   83 CONTINUE                                                          31
*              DRAW BORDER OF PAGE
      CALL INPLOT (0.,8.5,0.,11.,8.5,11.)                               33
      CALL PLOT(0.,11.,2)                                               34
      CALL PLOT(8.5,11.,2)                                              35
      CALL PLOT(8.5,0.,2)                                               36
      CALL PLOT(0.,0.,2)                                                37
      CALL PLOT(1.75,1.2,3)                                             38
      CALL PLOT(1.75,10.40,2)                                           39
      CALL PLOT(1.75,8.7,3)                                             40
*              LNEPLT PLOTS ONE ARRAY AND THEN REPOSITIONS THE PEN DOWNWARD
*              ACCORDING TO LAST ARGUMENT
*              PLOT TARGET PHASE
```

```
      CALL LNEPLT (-1.,YYYYY,-3.2,3.2,CELPHA,1,KTAREL,-11.4)        41
*             PLOT TARGET AMPLITUDE
      CALL LNEPLT (-1.,YYYYY,0.,TARG,CELAMP,1,KTAREL,-1.25*TARG)     42
*             PLOT RETURN PHASE
      CALL LNEPLT (XXXXX,PPP,-3.2,3.2,REPHA,LENUNC,NUMB,-11.4)      43
*             PLOT RETURN AMPLITUDE
      CALL LNEPLT (XXXXX,PPP,0.,VRET,REAMP,LENUNC,NUMB,-1.25*VRET)  44
*             PLOT PROCESSED IMAGE
      CALL LNEPLT (XXX,PPPPP,0.,VIMAG,AMAP,LENREF,L4,0.)            45
      PPPPP = 2.*PPPPP                                              46
      GGG = -1./1/1.4*VIMAG                                         47
      CALL PLOT (PPPPP,GGG,-2)                                      48
      RETURN                                                        49
      END                                                           50
```

23604 WORDS OF MEMORY USED BY THIS COMPILATION

```
      SUBROUTINE LNEPLT (XMIN,XMAX,YMIN,YMAX,ARRAY,I1,I2,DOWN)
*               THIS ROUTINE PLOTS ONE ARRAY AND THEN REPOSITIONS THE PEN
*               FOR THE NEXT PLOT, IT IS A SUBROUTINE OF OURPLT
*               XMIN = MINIMUM X COORDINATE TO BE PLOTTED
*               XMAX = MAXIMUM X COORDINATE TO BE PLOTTED
*               YMIN = MINIMUM Y COORDINATE TO BE PLOTTED
*               YMAX = MAXIMUM Y COORDINATE TO BE PLOTTED
*               ARRAY = ARRAY TO BE PLOTTED
*               I1 = FIRST ELEMENT OF ARRAY TO BE PLOTTED
*               I2 = LAST ELEMENT OF ARRAY TO BE PLOTTED
*               DOWN = DOWNWARD LENGTH FOR PEN TO BE MOVED AFTER COMPLETING
*               THE PLOT (REPOSITIONS FOR NEXT PLOT)
      DIMENSION ARRAY(1)
*               SET LIMITS FOR PLOT
      CALL INPLOT (XMIN,XMAX,YMIN,YMAX,6.,1,4)
      A = I1                                                        2
      CALL PLOT (A,ARRAY(I1),3)                                     3
*               PLOT ARRAY
      DO 99 I=I1,I2                                                 4
      R1 = I                                                        5
      R2 = I + 1                                                    6
      CALL PLOT(R1,ARRAY(I),2)                                      7
      CALL PLOT(R2,ARRAY(I),2)                                      8
   99 CONTINUE                                                      9
      CALL PLOT(XMAX,0.,3)                                         11
*               DRAW X AXIS
      CALL PLOT(XMIN,0.,2)                                         12
      CALL PLOT(XMIN,0.,3)                                         13
*               REPOSITION DOWNWARD FOR NEXT PLOT ACCORDING TO DOWN
      CALL PLOT(XMIN,DOWN,-2)                                      14
      RETURN                                                       15
      END                                                          16
```

23702 WORDS OF MEMORY USED BY THIS COMPILATION

```
      SUBROUTINE MENVAR(KSAMPD,K1SAMP,ARRAY,HEAD,KFSAMP)
*              MENVAR PERFORMS STATISTICAL ANALYSIS ON AN ARRAY, IT USES
*              A T-TEST TO FIND 90 PERCENT CONFIDENCE INTERVAL
*              ARRAY = ARRAY TO BE ANLYZED
*              KSAMPD = SAMPLING RATE TO BE USED
*              K1SAMP = FIRST ELEMENT OF ARRAY TO BE USED
*              KFSAMP = LAST ELEMENT OF ARRAY TO BE USED
*              KDIM = DIMENSION OF ARRAY
      DIMENSION V1(2),V2(2),V3(2),ARRAY(1),HEAD(2)
      IF(KSAMPD.GT.0) GO TO 6
      WRITE(6,20) HEAD,KSAMPD,                                          4
   20 FORMAT(/ ' SAMPLING RATE FOR ANALYSIS OF ',2A6,' =',I5,           7
     2    ' IS IN ERROR, ANALYSIS NOT POSSIBLE')
      RETURN                                                            7
    6 N = (KFSAMP - K1SAMP + 1)/2                                       8
      L1 = K1SAMP                                                       9
      L2 = N + K1SAMP - 1                                               10
      DO 1 I=1,2                                                        11
      V1(I)=0.                                                          12
      V2(I)=0.                                                          13
      V3(I)=0.                                                          14
      AL3=0.                                                            15
      AL5=0.                                                            16
      DO 2 J=L1,L2,KSAMPD                                               17
      AL3=AL3+1.                                                        18
      V1(I)=ARRAY(J)+V1(I)                                              19
      V2(I)=V2(I)+ARRAY(J)**2                                           20
      L4=J+KSAMPD                                                       21
      IF(L4 .GT. L2) GO TO 2                                            22
      AL5=AL5+1.                                                        25
      V3(I)=V3(I)+ARRAY(J)*ARRAY(L4)                                    26
    2 CONTINUE                                                          27
      L1 = L2 + KSAMPD                                                  29
      L2=L1+N-1                                                         30
      V1(I)=V1(I)/AL3                                                   31
      V2(I)=V2(I)/AL3                                                   32
    1 V3(I)=V3(I)/AL5                                                   33
*              XBAR = MEAN VOLTAGE
      XBAR=(V1(1)+V1(2))/2.                                             35
      S2=(V1(1)**2+V1(2)**2)/2.-XBAR**2                                 36
      CLSIG=XBAR-6.314*SQRT(S2)                                         37
      CHSIG=XBAR+6.314*SQRT(S2)                                         38
      CLSIG2=CLSIG**2                                                   39
      CHSIG2=CHSIG**2                                                   40
*              SBAR2 = VOLTAGE VARIANCE
      SBAR2=((V2(1)-V1(1)**2)+(V2(2)-V1(2)**2))/2.                      41
      SS2=((V2(1)-V1(1)**2)**2+(V2(2)-V1(2)**2)**2)/2.-SBAR2**2         42
      CLNOI=SBAR2-6.314*SQRT(SS2)                                       43
      CHNOI=SBAR2+6.314*SQRT(SS2)                                       44
      STN=XBAR**2/SBAR2                                                 45
      STNL=CLSIG2/CHNOI                                                 46
```

```
      STNH = CHSIG2/CLNOI                                              47
*               BARCOV = STATISTICAL COVARIANCE
      BARCOV=((V3(1)-V1(1)**2)/(V2(1)-V1(1)**2)+                       48
     1(V3(2)-V1(2)**2)/(V2(2)-V1(2)**2))/2.
      SCOV2=(((V3(1)-V1(1)**2)/(V2(1)-V1(1)**2))**2+                   49
     1((V3( 2)-V1(2)**2)/(V2(2)-V1(2)**2))**2)/2.-BARCOV**2
      CLCOV=BARCOV-6.314*SQRT(SCOV2)                                   50
      CHCOV=BARCOV+6.314*SQRT(SCOV2)                                   51
      AL3 = AL3 + AL3                                                  52
      WRITE(6,10) HEAD,KSAMPD,AL3                                      53
   10 FORMAT(//' STATISTICAL ANALYSIS OF ',2A6,'SAMPLED EVERY ',I2,' ELE   56
     2MENTS' / ' A TOTAL OF ',F5.0,' SAMPLES WERE USED IN THIS ANALYSIS'
     3 //)
      WRITE(6,3) XBAR,CLSIG,CHSIG,SBAR2,CLNOI,CHNOI,BARCOV,CLCOV,      56
     2         CHCOV,STN,STNL,STNH
    3 FORMAT(//1X,27HTHE MEAN STAT IMAGE VOLTAGE,F10.3/                59
     11X,38H90 PERCENT CONFIDENCE INTERVAL IS FROM,F10.3,5X,2HTO,F10.3//
     21X,36HTHE MEAN STAT IMAGE VOLTAGE VARIANCE,F10.3/
     31X,38H90 PERCENT CONFIDENCE INTERVAL IS FROM,F10.3,5X,2HTO,F10.3//
     41X,44HTHE MEAN STAT COVARIANCE OF IMAGE SAMPLES IS,F10.3/
     51X,38H90 PERCENT CONFIDENCE INTERVAL IS FROM,F10.3,5X,2HTO,F10.3//
     6 ' MEAN VOLTAGE SQUARED / VOLTAGE VARIANCE = ',F10.3/
     71X,38H90 PERCENT CONFIDENCE INTERVAL IS FROM,F10.3,5X,2HTO,F10.3/)
      RETURN                                                           59
      END                                                              60

   23661 WORDS OF MEMORY USED BY THIS COMPILATION
```

SETS ADDRESS FOR A WRAPUP ROUTINE IN CASE OF AN ABORT

```
PREFACE

  PROGRAM BREAK    20
  COMMON LENGTH     0
  V COUNT BITS      5

PRIMARY SYMDEF   ENTRY

   SETWRP           0

SECONDARY SYMDEF  ENTRY

   BLOCK       LENGTH

   SYMREF

 1 TWRAP
END OF BINARY CARD 00000001
```

SETS ADDRESS FOR A WRAPUP ROUTINE IN CASE OF AN ABORT

```
                                     1          TTL      S U B R O U T I N E  S E T W R P
                                     2          TTLS     SETS ADDRESS FOR A WRAPUP ROUTINE IN CASE OF AN ABORT
                                     3          SYMDEF   SETWRP
                           000000    4 SETWRP   SAVE
       000000  000002710000    010
       000001  000015630000    010
       000002  000015754000    010
       000003  000015741000    010
       000004  000027 2210 00  003   5          LDX1     23                  GET FORTRAN WRAPUP ADDRESS
       000005  000014 7410 00  010   6          STX1     GOTO                SAVE IT IN GOTO
       000006  000011 6210 00  010   7          EAX1     WRAPUP              GET MY WRAPUP ADDRESS
       000007  000027 7410 00  003   8          STX1     23                  PUT IT IN WORD 23
       000010  000001710000    010   9          RETURN   SETWRP              WE'RE DONE GO BACK TO MAINLINE
                           000011   10 WRAPUP   NULL                         CONTROL IS TRANSFERED HERE IN CASE OF AN ABORT
       000011  010000701000    030  11          CALL     TWRAP               CALL TAPE ROUTINE TO CLOSE TAPE IF NECESSARY
       000012  000014710000    010
       000013  000015000013    010
       000014  000000 7100 00  003  12 GOTO     TRA      **                  THEN GO TO FORTRAN WRAPUP

    ERROR LINKAGE

       000015  000000000000    000
       000016  622563665147    000
     END OF BINARY CARD 00000002
                                    13          END
      20 IS THE NEXT AVAILABLE LOCATION.   GMAP VERSION  JMPA/080870 JMPB/080870 JMPC/080870
    THERE WERE    NO  WARNING FLAGS IN THE ABOVE ASSEMBLY
```

OCTAL	SYMBOL	REFERENCES BY ALTER NO.			
14	GOTO	12	6	12	
15	.E.L..		4	11	
0	SETWRP	4	3	4	9
1	TWRAP		11		
11	WRAPUP	10	7	10	

** 17723 WORDS OF MEMORY WERE USED BY GMAP FOR THIS ASSEMBLY.

BIBLIOGRAPHY

1. Cutrona, L. J., Leith, E. N., Palermo, C. J., and Porcello, L. J., "Optical data processing and filtering systems," IRE Trans. Inform. Theory, vol. IT-6, pp. 386-400, 1960.

2. Cutrona, L. J., Leith, E. N., Porcello, L. J., and Vivian, W. E., "On the application of coherent optical processing techniques to synthetic aperture radar," Proc. IEEE, vol. 57, pp. 1026-1032, 1966.

3. Harger, R. O., "On processing optical images propagated through the atmosphere," IEEE Trans. Aerospace Electron. Systems, vol. AES-3, pp. 819-828, 1967.

4. Leith, E. N., and Upatnieks, J., "Reconstructed wave-fronts and communication theory,", J. Opt. Soc. Am., vol. 52, pp. 1123-1130, 1962.

5. Vanderlugt, A., "Operational notation for the analysis and synthesis of optical data processing systems," Proc. IEEE, vol. 54, pp. 1055-1063, 1966.

6. Harzer, R. O., Synthetic Aperture Radar Systems Theory and Design, Academic Press, New York, 1970.

7. Schwarz, D. E., Simonett, D. S., Jenks, G. F., Ratzlaff, J., "The construction of thematic land use maps with spacecraft photography," University of Kansas, CRES Rpt 117-4, May 1969.

8. Schwartz, M., Information Transmission Modulation and Noise, McGraw-Hill, New York, p. 168, 1959.

9. Klauder, J. R., Price, A. C., Darlington, S., and Alberscheim, W. J., "The theory and design of chirp radars," Bell System Tech. J., vol. J-39, pp. 745-808, July 1960.

10. Brown, W. M., "Synthetic aperture radar," IEEE Trans. Aerospace Electron. Systems, vol. AES-3, no. 2, March 1967.

11. Khambata, A. J., Introduction to Large Scale Integration, Willey-Interscience, 1969.

12. Gabor, D., "Theory of communication," J. Inst. Elec. Engrs. (London), pt. III, vol. 93, pp. 429-457, 1946.

13. Rihaczek, A. W., Principles of High Resolution Radar, McGraw-Hill, New York, 1969.

14. Woodward, P. M., Probability of Information Theory, with Applications to Radar, Pergamon Press, New York, 1953.

15. Beckmann, P., Spizzichina, A., The Scattering of Electromagnetic Waves from Rough Surfaces, MacMillan Co., New York, 1963.

16. Waite, W. P., "Broad-Spectrum Electromagnetic Backscatter," Ph.D. Thesis, University of Kansas, 1970.

17. Thomann, G. C., "Panchromatic Illumination for Radar; Acoustic Simulation of Panchromatic Radar," Ph.D. Thesis, University of Kansas, 1970.

18. Gerchberg, R. W., Haralick, R. M., "Correlated averaging to enhance radar imagery," CRES Tech. Memo. 118-15, University of Kansas, 1970.

19. Rice, S. O., "Mathematical analysis of random noise," Bell System Tech. J., vol. 23, pp. 282-332, 1944.

20. Moore, R. K. and Rouse, J. W., "Electronic processing for synthetic aperture array radars," Proceedings IEEE, vol. 55, no. 2, February 1967.

21. Haralick, R. M., Computer program PITCHR, on file at the University of Kansas Computer Center, 1969.

22. Lynch, Jerry, Private Communication, Texas Instruments, (Houston Facility)

23. Suran, J. J., "A perspective on integrated electronics," IEEE Spectrum, January 1970.

24. Von Neumann, J., Goldstine, H. H., "On the principles of large-scale computing machines," May 15, 1946, in John Von Neumann Collected Works, vol. 5, Macmillan Book, New York, 1963.

25. Field, S. W., "Semiconductor memories at a glance: What's here now and what's on the way," Electronics, vol. 43, no. 5, October 1969.

26. Stehlin, R. A., "The realization of 1.0 volt multivibrators using complementary integrated circuits," IEEE Journal of Solid-State Circuits, vol. SC-4, no. 5, October 1969.